

Borland® StarTeam® 2009

Starteam Cross-Platform Client Help

Borland®

Borland Software Corporation
8310 N Capital of Texas Hwy, Bldg 2, Ste 100
Austin, Texas 78731 USA
www.borland.com

Borland Software Corporation may have patents and/or pending patent applications covering subject matter in this document. Please refer to the product CD or the About dialog box for the list of applicable patents. The furnishing of this document does not give you any license to these patents.

Copyright © 1995–2009 Borland Software Corporation and/or its subsidiaries. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. All other marks are the property of their respective owners.

June 2009
PDF

StarTeam Cross-Platform Client

Getting Started	17
Introduction	18
Installing StarTeam	19
License Overview	20
About Source Control	22
StarTeam Product Overview	23
Standard StarTeam Architecture Overview	28
What's New in StarTeam 2009	30
New Features in StarTeam 2009 Server	31
New Features in the StarTeam 2009 Cross-Platform Client	33
New Features in View Compare/Merge	36
Borland StarTeam 2009 Web Client	41
New Features in Other StarTeam 2009 Components and Products	42
Help on Help	44
StarTeam Overview	45
Where to Find Documentation for Each Product	46
User Roles and StarTeam Documentation	49
StarTeam Basics	50
Containers	51
Artifacts	53
Artifacts Versus Items	55
Folders	56
Files	59
Change Requests	61
Requirements	66
Tasks	67
Topics	68
Links: Internal and External	69
Labels	70
Branching, Merging and Dot Notation	72
Sharing and "Cheap Copies"	76
Promotion States	78
Testing and Reporting	79
Testing and Reporting Overview	80
Change Requests	82
Queries	87
Reports	88
Charts	90
Filters	91
Fields	92
Data Export with Datamart	93
Cross-Server Configuration/Project Searches with Borland Search	94
Tour of the UI	95
Cross-Platform Client	96
Cross-Platform Client Overview	97
Change Perspective UI	102
Chart Window	107
View Compare/Merge (VCM) UI	109
View Compare/Merge	110
View Compare/Merge Status Icons	113
File Compare/Merge UI	115
Toolbar Utility	122
Concepts	124

Project Configuration	125
Overview of Projects	126
Keeping Projects Autonomous	127
Handling Cross-Project File Dependencies	129
Supporting Cross-Project Activities	131
View Configuration and Management	132
Overview of Views	133
Understanding View Types	136
View Roles	138
Proper Use of Views	142
Change Management within a View	143
Overview of Folders and Paths	146
Promotion States	148
Personal Options	151
Understanding Default and Alternate Working Folders	154
Check-in and Check-out Operations	155
Check-in and Check-out Overview	156
EOL Conversion Handling Overview	158
Achieving Consistent Check-ins and Check-outs	159
Bulk Check-out Utility for Large Numbers of Files	161
Labels	163
Labels	164
View Labels	166
Branching Options	168
Understanding Branching	169
Overview of Branching Options	170
Branching Behavior of Items	172
Effects on Change Requests When Branched, Moved, and Shared	174
Floating Items	175
References to Folders and Items	178
References Overview	179
Understanding References	180
References Created by Branching Views	184
References Created by Adding Items to Views	186
References Created by Manually Sharing Objects	189
References Created by Moving Objects	191
Process Items, Process Links, and Process Tasks	193
Process Items Overview	194
Process Items and Process Links	197
Process Rules	201
Process Tasks and Enhanced Process Links	203
Active Process Items	205
Comparing and Merging Files, Folders, and Views	206
Overview of File Compare/Merge	207
Overview of View Compare/Merge (VCM)	209
Change Packages Overview	215
Advantages of Change Packages	219
View Compare/Merge Session Perspectives	220
View Merge Type Scenarios	225
VCM Merge Types, Rules, and Scenarios	229
Access Rights Required for StarTeam VCM Users	233
Tips for Successful VCM Sessions	238
View Compare/Merge Action Decisions	240
View Compare/Merge Actions	241
Conditions for VCM Action Decisions	243

VCM Table of Action Decisions	245
Understanding VCM Difference Types	250
How VCM Handles Target Folders with Floating Shares	256
View Compare-only Difference Table	259
Atomic Check-ins	263
Quick Access to Projects and Items	264
Procedures	266
Logging On and Off	268
Logging Off	269
Logging On to Servers with the Toolbar Utility	270
Logging On to the Server and Starting a Project	271
Configuring Your Client	273
Adding Server Configurations	274
Adding Tools to the Toolbar Utility	276
Changing the Default Language and Locale	278
Changing Your Password	279
Configuring an Alternate Editor, Merge, or Comparison Utility	280
Configuring the Display Order of Component Tabs in the Client	281
Controlling How File Status Information is Stored	282
Customizing Personal Options	283
Customizing the Detail Pane	284
Customizing the Toolbar Utility	285
Adding Tools to the Toolbar Utility	286
Modifying Tool Properties in the Toolbar Utility	288
Deleting Tools from the Toolbar Utility	289
Refreshing the Toolbar Utility	290
Deleting Tools from the Toolbar Utility	291
Displaying Additional Fields	292
Displaying and Customizing Logging Options	293
Displaying Notifications in the Client	294
Editing your Account Information	296
Modifying Tool Properties in the Toolbar Utility	297
Refreshing the Toolbar Utility	298
Sorting and Grouping Data	299
Using the Toolbar Utility	301
Adding Tools to the Toolbar Utility	302
Customizing the Toolbar Utility	304
Adding Tools to the Toolbar Utility	305
Modifying Tool Properties in the Toolbar Utility	307
Deleting Tools from the Toolbar Utility	308
Refreshing the Toolbar Utility	309
Deleting Tools from the Toolbar Utility	310
Logging On to Servers with the Toolbar Utility	311
Modifying Tool Properties in the Toolbar Utility	312
Opening and Closing the Toolbar Utility	313
Refreshing the Toolbar Utility	314
Starting Applications with the Toolbar Utility	315
Managing Projects	316
Assigning Access Rights to Projects	318
Changing Project Names or Descriptions	319
Configuring Projects to use APES	320
Configuring Server Configurations to use Enhanced Process Links	321
Creating Projects	322
Creating Reports	323
Deleting Projects	324

Demoting View Labels	325
Displaying Location References	326
Enabling Keyword Expansion	328
Enabling the use of Enhanced Process Links	329
Establishing Process Rules for Projects	330
Opening Existing Projects	332
Opening Projects with Shortcuts	333
Promoting View Labels	334
Requiring Exclusive Locks for Check-ins	335
Requiring Revision Comments	336
Saving Projects as Shortcuts	337
Setting Active Process Items	338
Setting Folder Component Access Rights	339
Switching Views	340
Viewing Connection Properties	341
Viewing or Modifying Project Properties	342
Managing Views	343
Changing a View's Default and Alternate Working Folders	344
Copying View Labels	345
Creating and Configuring Views	347
Creating View Labels	349
Deleting Views	351
Modifying View Names or Descriptions	352
Refreshing Views	353
Reviewing or Modifying View Properties	354
Rolling Back the Current View Configuration	355
Branching Operations	357
Configuring a View to Display Non-Branched Files	358
Configuring the Branching Behavior of Shared Items	359
Creating a Branching View	360
Reviewing or Changing Branching Behavior	361
Comparing and Merging Views	362
Changing Access Rights for a Change Package	365
Changing the Default Merge Type Options and Actions	366
Changing the View Compare/Merge Session Working Folder	367
Changing View Compare/Merge Actions	368
Changing View Compare/Merge Perspectives	369
Cloning a New VCM Merge Type From One Server Configuration to Another	370
Committing View Compare/Merge Change Session	371
Comparing a Source and Target File in View Compare/Merge	373
Comparing a Target File with the Proposed Merge Results	375
Comparing and Merging Views	376
Preparing Your Files for a View Compare/Merge Session	377
Comparing and Merging Views	378
Resolving Merge Conflicts in View Compare/Merge	380
Changing View Compare/Merge Actions	382
Testing View Compare/Merge Changes	383
Viewing and Printing a View Compare/Merge Difference Report	384
Comparing and Merging Views	385
Comparing Selected Items in the Source and Target View	387
Creating a Custom VCM Merge Type	389
Creating a View Compare/Merge Change Package	391
Comparing and Merging Views	392
Saving a VCM Session to a Change Package	393
Deleting a Change Package	394

Deleting a View Compare/Merge Session	395
Editing a Change Package	396
Editing Change Package Properties	397
Exporting a View Compare/Merge Session	398
Filtering Items in a View Compare/Merge Session	399
Finding and Breaking Locks on Items with Differences	401
Merging Changes Between Any Two Related Views (Replicate)	403
Merging Changes from a Child View to a Parent View (Promote)	405
Merging Changes from a Parent View to a Child View (Rebase)	407
Naming a Change Package	409
Opening a Change Package in a VCM Session	410
Opening a Change Package in Read-Only Mode for Review	411
Opening a Saved or Exported View Compare/Merge Session	412
Preparing Your Files for a View Compare/Merge Session	413
Replaying a Change Package to Another View	414
Resolving File Differences from a View Compare/Merge Session	416
Resolving Merge Conflicts in View Compare/Merge	417
Restarting a VCM Session	419
Running View Compare/Merge from the Command-line	420
Saving a VCM Session to a Change Package	421
Setting Locking Options for a View Compare/Merge Session	422
Specifying the Initial Perspective for a View Compare/Merge Session	423
Testing View Compare/Merge Changes	424
Using Filters and Queries to Navigate Change Packages	425
Using Process Items to Merge Related Files	426
Viewing and Printing a VCM Session Summary	428
Viewing and Printing a View Compare/Merge Difference Report	429
Viewing Change Package Details, History and Labels	430
Viewing Changes in a Change Package	431
Viewing the Change Perspective	432
Comparing and Merging Files and Folders	433
Applying Character Encoding to Files	434
Comparing a Local File with a Repository File	435
Comparing and Merging Files	436
Comparing a Local File with a Repository File	437
Editing Files in a File Compare/Merge Session	438
Generating Reports from a File Compare/Merge Session	439
Saving Files Modified in a File Compare/Merge Session	440
Comparing Folders	442
Comparing Historical File Contents	444
Comparing Images	445
Comparing Two Local Files	447
Editing Files in a File Compare/Merge Session	448
Generating Reports from a File Compare/Merge Session	449
Merging a Local File with the Tip Revision	450
Merging Folders	452
Merging Two Local Files	454
Saving Files Modified in a File Compare/Merge Session	456
Setting File Compare/Merge Options	458
Working with Folders and Items	459
Adding Folders to Views	461
Attaching Labels to Folders	463
Attaching Labels to Items	465
Changing a View's Default and Alternate Working Folders	468
Changing Name or Description of Folders and Items	469

Comparing Properties of Two Non-file Items	470
Configuring (Rolling Back) Folders and Items	471
Controlling EOL Characters	473
Creating a Working Folder	474
Creating Reports	475
Customizing Link Item Properties	476
Deleting Folders and Items	477
Displaying Item Details	478
Displaying Location References	479
Emailing Item Properties	481
Excluding Files and Folders from a Project	482
Finding Items	483
Hiding Folders and Files	484
Highlighting Items of Interest	485
Linking Items Internally or Externally	486
Locking and Unlocking Items	488
Marking Items Read or Unread	490
Merging Folders	491
Moving Folders or Items	493
Opening a Local Folder from StarTeam	495
Renaming Files	496
Restoring Folder Selection on Tab Change	497
Selecting Referenced Items in Other Views	498
Sharing Folders or Items	499
Sorting and Grouping Data	501
Viewing or Modifying Item Properties	503
Viewing Previous File Revisions	504
Managing Files	505
Checking In Files	507
Checking Out Files	509
Adding Files to Projects	511
Checking Out Files in Batches	512
Checking Out Historical Versions of Files	514
Controlling How File Status Information is Stored	515
Deleting Folders and Items	516
Displaying Item Details	517
Editing Check-in Comments	518
Enabling Concurrent File Editing	519
Excluding Files and Folders from a Project	520
Finding Files Linked to Active Process Items	521
Hiding Folders and Files	522
Locking and Unlocking Items	523
Marking Unlocked Files Read-only	525
Monitoring Check-out Statistics using Cache Agent	526
Opening a Local Folder from StarTeam	527
Opening and Editing Files	528
Renaming Files	530
Selecting Linked Files	531
Setting File Storage Options	532
Setting the File Executable Bit for UNIX	533
Using the File Annotation Tool	534
Viewing or Modifying Item Properties	536
Viewing Previous File Revisions	537
Working with Change Requests	538
Adding Change Request Comments	540

Assigning Change Requests	541
Closing Verified Change Requests	542
Creating Change Requests	543
Specifying Change Request Summary Information	544
Specifying Change Request Descriptions	545
Specifying Change Request Solutions	546
Modifying Custom Options for Change Requests	547
Including Change Request Attachments	548
Adding Change Request Comments	549
Customizing Change Request Filters	550
Customizing Change Request Reports	551
Showing Fields in a Change Request	552
Selecting Change Requests Using a Query	553
Sorting and Grouping Change Requests	554
Including Change Request Attachments	555
Linking Items Internally or Externally	556
Locking and Unlocking Items	558
Modifying Custom Options for Change Requests	560
Resolve Open Change Requests	561
Reviewing Linked Change Requests	563
Selecting Change Requests Using a Query	564
Showing Fields in a Change Request	565
Sorting and Grouping Change Requests	566
Specifying Change Request Descriptions	567
Specifying Change Request Solutions	568
Specifying Change Request Summary Information	569
Verifying Resolved Change Requests	570
Working with Change Request Lists	572
Using Requirements	574
Creating Requirements	575
Marking Item Threads Read or Unread	577
Using Tasks	578
Add Notes to Tasks	579
Adding Comments to Task Revisions	580
Assigning Task Resources	581
Creating Tasks	583
Customizing Tasks	585
Marking Item Threads Read or Unread	586
Scoping Tasks	587
Working with Attachments in Tasks	588
Working with Work Records in Tasks	590
Using Topics	592
Creating Topics	593
Marking Item Threads Read or Unread	595
Responding To Topics	596
Viewing the Audit Log	598
Filtering Audit Log Entries	599
Searching for Log Entries	600
Sending Log Entries Through E-mail	601
Sorting Audit Log Entries	602
Managing Labels and Promotion States	603
Attaching Labels to Folders	604
Attaching Labels to Items	606
Configuring or Viewing Label Properties	609
Configuring Promotion States	611

Copying Revision Labels	613
Copying View Labels	615
Creating Revision Labels	617
Creating View Labels	618
Deleting Labels	620
Demoting View Labels	621
Detaching a Label from a Rolled-back View	622
Detaching a Label from a Specific Revision	623
Detaching Labels from Folders	624
Detaching Labels from Items	625
Freezing or Unfreezing Labels	626
Promoting View Labels	627
Reviewing and Moving Labels	628
Sorting Labels Alphabetically	630
Linking and Unlinking Items	631
Checking Linked Files In and Out	632
Creating External Links	633
Customizing Link Item Properties	635
Customizing Link Properties	636
Deleting Links	637
Displaying Only Enhanced Process Links	638
Enabling the use of Enhanced Process Links	639
Filtering Process Tasks From Other Tasks	640
Finding Files Linked to Active Process Items	641
Linking Files to Process Items	642
Linking Items Internally or Externally	644
Linking Specific Revisions	646
Reviewing Linked Change Requests	648
Selecting Linked Files	649
Viewing Links	650
Querying Data	651
Applying Queries	652
Copying Queries	653
Creating Queries	654
Creating “Me” Queries	656
Deleting Queries	657
Editing Queries	658
Selecting Change Requests Using a Query	659
Filtering Data	660
Applying Predefined Filters	661
Copying Filters	662
Creating Filters	664
Customizing Change Request Filters	666
Deleting Filters	667
Editing Filters	668
Filtering Process Tasks From Other Tasks	669
Resetting Filters	670
Showing Fields in a Change Request	671
Sorting and Grouping Change Requests	672
Sorting and Grouping Data	673
Specifying Process Rules and Process Items	675
Displaying Only Enhanced Process Links	676
Establishing Process Rules for Projects	677
Filtering Process Tasks From Other Tasks	679
Finding Files Linked to Active Process Items	680

Linking Files to Process Items	681
Promoting File Changes Into Baselines	683
Viewing Links	684
Viewing Process Rules	685
Accessing Projects and Items with Shortcuts and URLs	686
Copying and Opening URLs	687
Creating Shortcuts	688
Working with Revisions	689
Checking Out Historical Versions of Files	690
Comparing Historical File Contents	691
Comparing Historical Properties	692
Comparing Properties	693
Copying Revision Labels	694
Creating Revision Labels	696
Detaching a Label from a Rolled-back View	697
Detaching a Label from a Specific Revision	698
Displaying Historical Properties	699
Editing Check-in Comments	700
Linking Specific Revisions	701
Viewing Previous File Revisions	703
Working with Properties	704
Comparing Historical Properties	705
Comparing Properties	706
Configuring or Viewing Label Properties	707
Customizing Link Item Properties	709
Customizing Link Properties	710
Displaying Historical Properties	711
Emailing Item Properties	712
Using Embedded Property Editors	713
Viewing or Modifying Item Properties	715
Creating Reports and Exporting Data	716
Configuring the Report Output Path	717
Creating Reports	718
Customizing Change Request Reports	719
Showing Fields in a Change Request	720
Selecting Change Requests Using a Query	721
Sorting and Grouping Change Requests	722
Customizing Fields Used In Report Templates	723
Customizing Report Templates	724
Exporting Data	725
Generating Reports from a File Compare/Merge Session	726
Printing Reports	727
Customizing Reports	728
Customizing Change Request Reports	729
Showing Fields in a Change Request	730
Selecting Change Requests Using a Query	731
Sorting and Grouping Change Requests	732
Customizing Compare and Merge Reports	733
Customizing Fields Used In Report Templates	734
Customizing Report Templates	735
Creating Charts	736
Choosing the Chart Type	737
Configuring Chart Colors	738
Customizing Chart Titles	739
Exporting a Chart as an Image	740

Generating Correlation Charts	741
Generating Distribution Charts	742
Generating Simple Charts	743
Generating Time-series Charts	744
Viewing Charts	745
Accessing Projects and Items with Shortcuts and URLs	747
Copying and Opening URLs	748
Creating Shortcuts	749
Reference	750
Access Rights and Privileges	752
Group Privileges	753
Server Access Rights	754
Project Access Rights	755
View Access Rights	756
Folder Access Rights	757
Child Folder Access Rights	758
File Access Rights	760
Generic Item Access Rights	762
Promotion State Access Rights	763
Component Access Rights	764
Component-level Filter Access Rights	765
Individual Filter Access Rights	766
Component-level Query Access Rights	767
Individual Query Access Rights	768
Change Package Access Rights	769
External Link Access Rights	771
File Compare/Merge Reference	772
File Compare/Merge Actions	773
File Compare/Merge Keyboard Shortcuts	777
File Compare/Merge Options	779
General (File Compare/Merge Options)	780
File Comparison (File Compare/Merge Options)	782
Folder Comparison (File Compare/Merge Options)	789
View Compare/Merge	791
View Compare/Merge Wizard	792
View Compare/Merge Wizard: Select Compare/Merge Type	793
View Compare/Merge Wizard: Include Items	794
View Compare/Merge Wizard: Include Selected Items	795
View Compare/Merge Wizard: Select Source View	796
View Compare/Merge Wizard: Select Target View	797
View Compare/Merge Wizard: Select Source Configuration	798
View Compare/Merge Wizard: Select Target Configuration	799
View Compare/Merge Wizard: Set Options	800
View Compare/Merge Wizard: Exclude Properties	802
View Compare/Merge Session Filters	803
Change Package Properties	804
Versioning Object Properties	806
View Compare/Merge Terminology	808
Cross-Platform Client Menus	812
Project Menu	813
View Menu	814
Folder Tree Menu	815
Folder Menu	816
File Menu	819
Change Request Menu	822

Requirement Menu	825
Task Menu	828
Topic Menu	831
Audit Menu	834
Tools Menu	836
Window Menu	837
Help Menu	838
Customization	839
Email Customization Reference	840
Change Request Configuration File	841
Change Request Message Template Syntax	843
Detail Pane Customization Reference	845
Sample Folder Template	846
Sample File Template	848
Sample Change Request Template	850
Sample Task Template	853
Sample Topic Template	856
Sample Requirement Template	858
Sample Change Package Template	860
Keyboard Shortcuts	862
Client Keyboard Shortcuts	863
File Compare/Merge Keyboard Shortcuts	864
Server Administration Keyboard Shortcuts	866
Project, Folder, and Item Properties	867
Folder Properties	868
File Properties	871
Change Request Properties	873
Requirement Properties	875
Task Properties	878
Topic Properties	881
Project Properties Dialog Box	883
Name (Project Properties Dialog Box)	884
Options (Project Properties Dialog Box)	885
Default Types (Project Properties Dialog Box)	886
Process Rules (Project Properties Dialog Box)	887
Editors (Project Properties Dialog Box)	889
Change Package Properties	890
Versioning Object Properties	892
Change Requests	894
Default and Required Change Request Fields	895
Commonly Used Change Request Abbreviations	898
Charts	899
Chart Types	900
Default Chart Views and Zoom/Rotate Capabilities	902
File Chart Fields	903
Change Request Chart Fields	904
Requirement Chart Fields	905
Task Chart Fields	906
Topic Chart Fields	907
Audit Chart Fields	908
Exporting	909
Export Dialog Box Options	910
Fields	911
File Fields	912
Change Request Fields	920

Requirement Fields	928
Task Fields	934
Topic Fields	941
Folder Fields	946
Audit Fields	951
Relational Operators Used in Queries	955
File Status	957
File Status Information	958
Effects of Status on Check-ins and Check-outs	960
Filters Reference Topics	962
File Filters	963
Change Request Filters	964
Requirement Filters	965
Folder Filters	966
Task Filters	967
Topic Filters	968
Audit Filters	969
View Compare/Merge Session Filters	970
Links	971
Link Tab	972
Queries	974
Predefined Queries	975
Relational Operators Used in Queries	976
New Query Options	978
Queries Options	980
Reports	981
Available Audit Reports	982
Available Change Request Reports	983
Available File Reports	984
Available Folder Reports	985
Available Requirement Reports	986
Available Task Reports	987
Available Topic Reports	988
Report Templates	989
Personal Options Settings	992
Workspace Options (Personal Options Dialog Box)	993
StarTeamMPX Options (Personal Options Dialog Box)	996
File Options (Personal Options Dialog Box)	998
Change Request Options (Personal Options Dialog Box)	1001
Requirements Options (Personal Options Dialog Box)	1003
Task Options (Personal Options Dialog Box)	1005
Topic Options (Personal Options Dialog Box)	1007
Folder Options (Personal Options Dialog Box)	1009
Views	1011
View Type Options and Settings	1012
View Configuration Options	1015
Audit Log Events	1016
Table of Common Operations	1017
Table of StarTeam Keywords	1019

StarTeam Cross-Platform Client

This section explains the concepts and procedures related to using the Cross-Platform Client

In This Section

[Getting Started](#)

This section contains basic conceptual topics related to software change management.

[Concepts](#)

This section contains all the conceptual topics.

[Procedures](#)

This section contains all the tasks associated with administering and using StarTeam.

[Reference](#)

This section contains all reference topics.

Getting Started

Thank you for choosing StarTeam!

This section contains basic conceptual topics related to software change management.

In This Section

[Introduction](#)

This section provides introductory information about StarTeam.

[What's New in StarTeam 2009](#)

This section contains 'What's New' information for this release.

[Help on Help](#)

This section describes the StarTeam Help system.

[StarTeam Basics](#)

The topics in this section provide an overview of basic StarTeam concepts.

[Testing and Reporting](#)

This section contains conceptual topics related to testing and reporting.

[Tour of the UI](#)

This section contains conceptual topics describing the StarTeam user interface.

Introduction

This section provides introductory information about StarTeam.

In This Section

[Installing StarTeam](#)

Link to the PDF file containing StarTeam installation procedures.

[License Overview](#)

This topic describes licensing options for StarTeam.

[About Source Control](#)

This topic describes source control at a high level.

[StarTeam Product Overview](#)

This topic describes the products that make up StarTeam.

[Standard StarTeam Architecture Overview](#)

This topic provides an overview of the standard StarTeam architecture.

Installing StarTeam

Installation instructions for installing StarTeam products can be found in *Installing StarTeam* . To view this document, choose [Start ▶ Programs ▶ Borland StarTeam ▶ StarTeam Cross-Platform Client 2009 ▶ Documentation ▶ Installation](#), or [Start ▶ Programs ▶ Borland StarTeam ▶ StarTeam Server 2009 ▶ Documentation ▶ Installation](#).

License Overview

This topic explains licensing for StarTeam, that is, the license package that you purchase and the different types of licenses available to determine how many users access StarTeam.

License Packages

StarTeam Server can be run as an Enterprise or Enterprise Advantage server, each of which has a different set of features. The features that a client can access on the server is determined by the license package that you purchase.

- ◆ Enterprise has all basic features including the Task component, the ability to customize properties for any component, and the Web Client
- ◆ Enterprise Advantage has all the Enterprise features plus the Requirement component, StarTeamMPX, and the alternate property editors that enable you to create custom forms and design workflow rules to control how all the items in a component move from state to state.
- ◆ Evaluation licenses are automatically installed and activated when you install the server. These licenses provide the features that you would get by using an Enterprise Advantage license and expire after a certain number of days.

If you change the registered license while a StarTeam project is open on a user's workstation, the licensing takes effect for that user by closing and reopening the project window. If you license a StarTeam Server as Enterprise after using an evaluation license (which is for the Enterprise Advantage edition) the feature set will change. For example, if you created requirements during the evaluation and license the server as anything other than Enterprise Advantage, the requirements tab will no longer display in the client.

Named User, Concurrent, and Borland License Types

Licenses also determine how many users can access StarTeam Server. Users can have either *named user* or *concurrent* licenses.

A named user license can be used only by the user who has been assigned that license. For example, if you have 5 named user licenses and 25 concurrent licenses, the 5 users who receive the named user licenses are guaranteed access to the server. No one else can use their licenses.

A concurrent license can be used by any user who does not have a named user license. For example, users without named user licenses receive concurrent licenses on a first-come, first-served basis. After all the concurrent licenses are in use, users attempting to log on are notified that there are no more licenses available at this time. They can try again later. Note that the Cross-Platform Client and the Web Client consume licenses separately.

When you first register the server, you enter one or two serial numbers: one for named user licenses and/or one for concurrent licenses. When using multiple serial numbers, they must all identify the same StarTeam edition (that is, Enterprise or Enterprise Advantage).

You can add more named user or concurrent licenses. StarTeam Server keeps track of the total number by summing the licenses supplied in each serial number or slip. This is referred to as stackable licensing.

You can add or import as many users as you choose, but access to the server is granted only to users with named user licenses or to users who receive concurrent licenses as they log on. If you have StarTeam named user licenses, you must assign them to specific users in the **User Manager** dialog (found in the Server Administration Window). Everyone else is assumed to have a StarTeam concurrent license.

If you have Borland licenses, users must be assigned to the correct slip in the User Manager dialog, regardless of their named or concurrent user status. An additional status, *Unassigned*, may be used instead of a slip.

The StarTeam Server Administrator is automatically assigned a named user license that cannot be removed. This is a "free" license that is not counted against the number of named user licenses you have available.

Using StarTeam Licensing

StarTeam Server can be licensed in either of two ways:

- ◆ StarTeam licensing, also referred to as native licensing, which is internal to the product
- ◆ Borland licensing available for use with license servers (BLS and FlexLM)

If StarTeam users attempt to access a server configuration that is managed by an unlicensed version of StarTeam Server, the tabs in the upper and lower panes of their StarTeam clients will not display.

Customers buy named Enterprise, concurrent Enterprise, named Enterprise Advantage, or concurrent Enterprise Advantage licenses for StarTeam Server. Any client can access any server as long as that server recognizes the user and has a license for that user. Customers usually choose just one method of licensing: StarTeam native licensing, Borland License Server licensing, or FlexLM licensing, but combinations can be supported. Licensing is handled after the installation, either by setting up a licensing server and putting "slip" files in the StarTeam Server's \licenses folder (a child of the server's installation folder) or by registering StarTeam native licenses using the StarTeam Server Administration tool.

Because StarTeam Server licenses are stackable, you can enter more than one license key so long as all the license keys are for the same edition (Enterprise or Enterprise Advantage). Be sure to delete the evaluation license before entering the first new license.

When you first register your server, you enter one or two serial numbers: one for named user licenses and/or one for concurrent licenses. When using multiple serial numbers, they must all identify the same StarTeam edition. You can add more named user or concurrent licenses. StarTeam Server keeps track of the total number by summing the licenses supplied in each serial number or slip. This is referred to as stackable licensing.

Related Concepts

[StarTeam Product Overview](#)

About Source Control

This topic describes source control at a high level, including basic information about source control and repositories.

Source Control Basics

Each source control system consists of one or more centralized repositories and a number of clients. A repository is a database that contains not only the actual data files, but also the structure of each project you define.

Most source control systems adhere to a concept of a logical project, within which files are stored, usually in one or more tree directory structures. A source control system project might contain one or many IDE-based projects in addition to other documents and artifacts. The system also enforces its own user authentication or, very often, takes advantage of the authentication provided by the underlying operating system. Doing so allows the source control system to maintain an audit trail or snapshot of updates to each file. By storing only the differences, the source control system can keep track of all changes with minimal storage requirements. When you want to see a complete copy of your file, the system performs a merge of the differences and presents you with a unified view. At the physical level, these differences are kept in separate files until you are ready to permanently merge your updates, at which time you can perform a commit action.

This approach allows you and other team members to work in parallel, simultaneously writing code for multiple shared projects, without the danger of an individual team member's code changes overwriting another's. Source control systems, in their most basic form, protect you from code conflicts and loss of early sources. Most source control systems give you the tools to manage code files with check-in and check-out capabilities, conflict reconciliation, and reporting capabilities. Most systems do not include logic conflict reconciliation or build management capabilities.

Commonly, source control systems only allow you to compare and merge revisions for text-based files, such as source code files, HTML documents, and XML documents. StarTeam stores binary files, such as images or compiled code, in the projects you place under control. You cannot, however, compare or merge revisions of binary files. If you need to do more than store and retrieve specific revisions of these types of files, you might consider creating a manual system to keep track of the changes made to such files.

Repository Basics

Source control systems store copies of source files and difference files in some form of database repository. In some systems, such as CVS or VSS, the repository is a logical structure that consists of a set of flat files and control files. In other systems, such as StarTeam, the repositories are instances of a particular database management system (DBMS) such as MS SQL Server or Oracle.

Repositories are typically stored on a remote server, which allows multiple users to connect, check files in and out, and perform other management tasks simultaneously.

With StarTeam, you create a *server configuration* to identify a repository for StarTeam projects. Each server configuration acquires its own set of projects as they are created. The Server can run any number of server configurations. Because each server configuration must use a database, you need to make sure that you establish connectivity not only with the server, but also with the database instance.

Related Concepts

[StarTeam Product Overview](#)

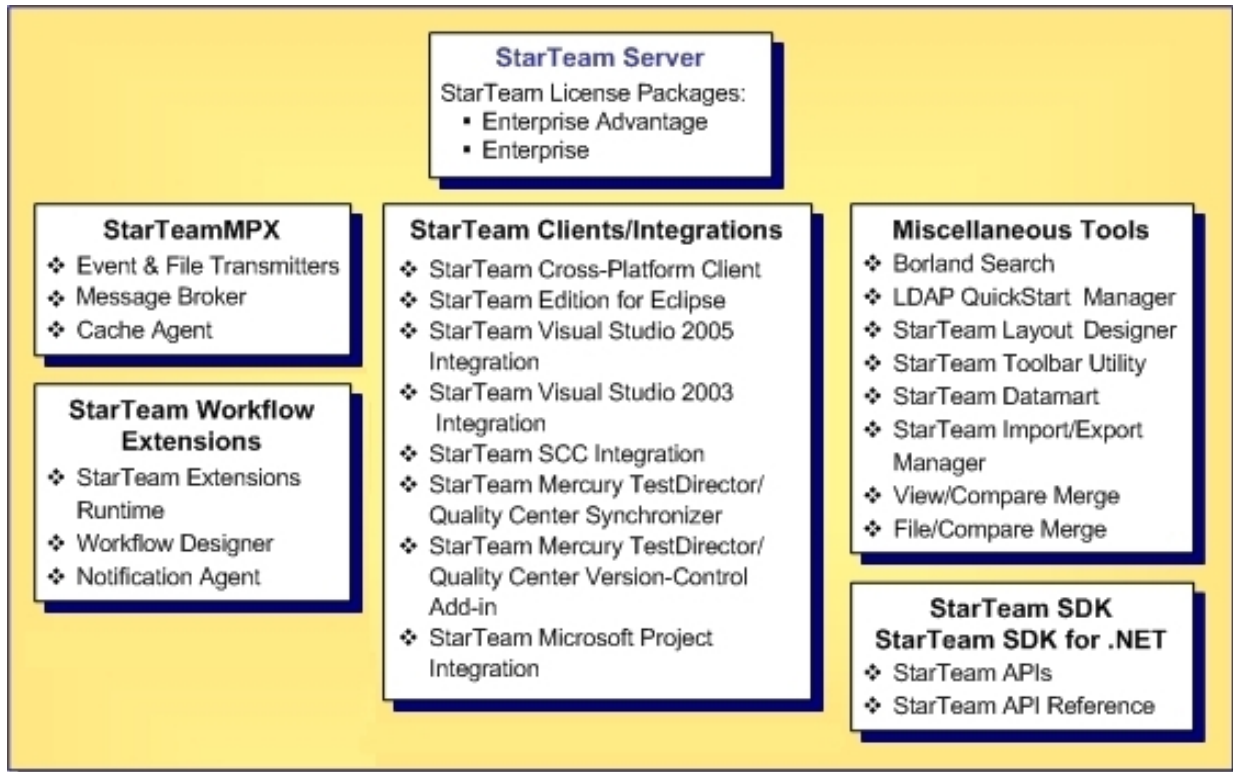
StarTeam Product Overview

This topic describes the products that make up StarTeam. Each product is described in the following sections.

The product descriptions in the sections that follow indicate if it is included in a particular licensing package. StarTeam is available in two licensing packages:

- ♦ **Enterprise:** StarTeam Enterprise provides a basic feature set, including the StarTeam Server, StarTeamMPX (Event Transmitter and Message Broker), the Cross-Platform Client, Web Client, LDAP Quick Start Manager, and the SDK. The requirements component is not available with this license; however, it does provide access to custom fields.
- ♦ **Enterprise Advantage:** StarTeam Enterprise Advantage has all the StarTeam Enterprise features plus the Requirement component, StarTeamMPX (Cache Agent and File Transmitter), and StarTeam Workflow Extensions which include alternate property editors (APEs) that enable you to create custom forms and design workflow rules to control how all the items in a component move from state to state. StarTeam Datamart is available for purchase.

StarTeam Products



The StarTeam family of products includes the StarTeam Server, Cross-Platform Client, Web Client, StarTeamMPX, StarTeam Extensions, tools and utilities to use with the clients and server, such as Borland Search, Borland LDAP QuickStart Manager, File Compare/Merge, and StarTeam Datamart, and a variety of integrations with third-party products, including integrations with Microsoft Visual Studio, Microsoft Project, and the Microsoft SCC Integration. Each product is described in more detail in the following sections.

StarTeam Server and client workstations are connected to maintain the repository, store changes made to files, and grant users access to project data.

StarTeam Server

Available for Windows and Linux.

StarTeam Server is a powerful tool that supports distributed development teams and mobile team members. It supports data in all languages that can be encoded in UTF-8. You can access the data managed by StarTeam Server using a variety of clients, such as the Cross-Platform Client or Web Client. Each client must have a user name and the correct access rights to access the selected server configuration (an instance of the StarTeam Server).

StarTeam clients use already familiar applications to access the server. For example, you can access the server from Internet Explorer using Web Client. If you use a StarTeam IDE integration, you can access StarTeam Server from IDE applications such as Microsoft Visual Studio and platforms such as Eclipse.

Access to StarTeam Server can be local or remote—via the Internet, intranet, or WAN. Built-in encryption enables you to work securely over public networks such as the Internet. Normally, you install StarTeam Server on a computer accessible to all team members. You then install StarTeam clients on team members' workstations.

StarTeamMPX

Available with Enterprise licenses: Event Transmitter and Message Broker.

Available with Enterprise Advantage licenses: All of StarTeamMPX Enterprise license features plus File Transmitter and Cache Agent.

This product is an addition to the StarTeam Server and must be installed separately. It uses advanced caching and publish/subscribe communication technology to improve the performance of StarTeam clients and extend the scalability of StarTeam Server. A Linux version of StarTeamMPX is also available.

StarTeam Workflow Extensions

Available with Enterprise Advantage licenses.

StarTeam Workflow Extensions enable you to create custom workflows for StarTeam components, such as change requests and tasks. You can customize the built-in workflow using alternate property editors (APEs), the Workflow Designer, and the Notification Agent.

Alternate Property Editors (APEs)	APEs are forms written in Java that replace the standard properties dialogs that come with each component (files, change requests, and so on) of the application.
Workflow Designer	StarTeam includes its own built-in workflow. If you intend to use your own custom workflow, you can use Workflow Designer to develop it. Workflow Designer outputs <i>item_type.Workflow.xml</i> files that formalize the steps in a workflow, specifies who will be notified in each step of the workflow or about exceptions and so on. Each *.Workflow.xml file can be used for an entire project or individual views within that project. The StarTeam Extensions workflow engine and Notification Agent read from the *.Workflow.xml files generated by Workflow Designer.
Notification Agent	Notification Agent monitors server configurations to determine the users that need to be notified about pending work and about exceptions that occur in the workflow process.

Cross-Platform Client

Available with both licenses.

First introduced in 2001, the Cross-Platform Client is a pure Java client that provides support of operating systems where a compatible JRE or JDK are available. As such, Cross-Platform Client is available for the Windows, Solaris,

and Linux operating systems. For the StarTeam release, the Cross-Platform Client has been given many quality enhancements.

StarTeam Edition for Eclipse

Available with both licenses.

StarTeam Edition for Eclipse allows you to share projects on StarTeam Server and projects in the Eclipse workspace, but it is much more than just a version control plug-in. This integration offers project teams a customizable solution providing requirements, task, and change management, defect tracking and threaded discussions tightly integrated within the Eclipse platform.

StarTeam Visual Studio Integration

The StarTeam Visual Studio Integration provides the StarTeam software configuration management capabilities tightly integrated with the Visual Studio development environment. Using this integration makes it possible for you to develop applications in the Visual Studio environment while simultaneously using the version control, change request, topic, task, and requirement component assets of StarTeam. The integration brings StarTeam main menu commands, context menu commands, and an embedded StarTeam client (providing much of the same look-and-feel as the full-featured Cross-Platform Client) to the Visual Studio development environment.

StarTeam Web Client

The new StarTeam Web Client is an intuitive web-based interface that many simultaneous users can use to connect to one or more StarTeam Servers to access projects and manage items. This initial release of the Web Client delivers a core feature set designed to meet the needs of users responsible for viewing, creating, and editing StarTeam change requests, requirements, tasks, and topics. Browsing files and a limited set of file operations are also available.

Note: You must have a StarTeam user license to use the Web Client.

StarTeam SCC Integration

Available with both licenses.

The StarTeam SCC Integration works with any application that uses the Microsoft Source Code Control (SCC) Application Programming Interface (API). This API, originally designed by Microsoft to allow applications to work with Microsoft Visual SourceSafe, enables you to perform version control operations, such as checking files in and out, using StarTeam as the SCC provider.

StarTeam Synchronizer for Mercury TestDirector for Quality Center

This product is available with both licenses.

StarTeam Synchronizer for Mercury TestDirector for Quality Center can ensure that the same data appears in Quality Center and a database used by StarTeam Server. The goal of the synchronization is to provide access to the latest information about defects, whether the defects are being processed from Quality Center or from StarTeam. You can use Quality Center to add defects, and you can use StarTeam to indicate that those defects have been fixed and vice versa. Team members do not need to be aware of where the defect was last processed. The latest data is available at all times, as long as the databases are synchronized frequently.

StarTeam Version-Control Add-in for Mercury TestDirector for Quality Center

Available with both licenses.

The StarTeam Version-Control Add-in for Mercury TestDirector for Quality Center enables you to place current and prior versions of Quality Center test plans under version control in the StarTeam repository. It supports both the Windows and Linux versions of StarTeam Server.

StarTeam Microsoft Project Integration

Available with both licenses.

The interoperation of the StarTeam Microsoft Project Integration and Microsoft Project make the jobs of both project planners and team members easier. Project planners use Microsoft Project to list the tasks that workers must perform. After exporting the tasks to StarTeam, they can gather information about the work accomplished by each team member in StarTeam — rather than communicating individually with each team member.

Borland Search

Available with Enterprise Advantage licenses.

Borland Search allows users to perform ad hoc queries across servers and projects. The query results reflect the access rights of the user logged on to Borland Search so information is shared across the organization without compromising security.

Borland LDAP QuickStart Manager

Available with both licenses.

Borland LDAP QuickStart Manager is a utility that allows you to import user information from a directory service or LDIF file into a CaliberRM or StarTeam Server. The imported user information is stored as user properties on each respective server.

StarTeam Layout Designer

Available with both licenses for the Cross-Platform Client and Web Client client.

The StarTeam Layout Designer provides the ability to customize forms within the application. Custom forms can be used to show custom properties, hide default properties that are not of interest to your organization, or rearrange the interface to more closely meet your organization's requirements.

StarTeam Toolbar Utility

The StarTeam Toolbar Utility (Toolbar) is a component of the StarTeam and CaliberRM products designed to make it easier for you to log on to multiple servers and to launch different programs. It automatically caches the user name and password used to log on to each StarTeam or CaliberRM server, reducing the number of times that you must enter your logon information. The Toolbar is initially populated with shortcuts for the tools of the StarTeam and CaliberRM products that are installed on your workstation. Because the Toolbar uses the standard Windows program shortcut feature, you can easily add any other program as a tool.

StarTeam Datamart

Available with Enterprise Advantage licenses. Can be purchased separately with Enterprise licenses.

StarTeam Datamart is a complementary product to the StarTeam Server. StarTeam Datamart uses the StarTeam SDK to communicate with the StarTeam Server to create a reporting database that you can use with popular third

party reporting applications such as Crystal Reports and Business Objects (reporting applications are not included with StarTeam Datamart). StarTeam Datamart extracts data from a StarTeam Server and places the data into a relational database, where reporting tools can access it. StarTeam Datamart can extract information from every project, every view in each project, every folder in each view, and every item in each folder, and labels, links, and history for each item. You can restrict extraction of data to a particular project and view or only extract certain tables.

StarTeam Import/Export Manager

Available for both licenses.

StarTeam Import/Export Manager is a set of utilities that allow you to copy a project from one StarTeam Server to another as a one-time necessity.

File Compare/Merge

File Compare/Merge is a graphical compare/merge tool delivered with the Cross-Platform Client. It enables you to compare a file dynamically with the file in the repository, and manually or automatically merge the content of the two files. File differences are highlighted in the File Compare/Merge panes using a configurable color scheme, and action buttons display in the highlighted areas to simplify the merging process.

View Compare/Merge

View Compare/Merge is a comprehensive tool for comparing and merging views available with the Cross-Platform Client. There are two versions of View Compare/Merge:

- ◆ Graphical: Provides interactive comparison and merging with per-item and per-folder interaction, allowing you to carefully control which items are compared and how each difference is resolved.
- ◆ Command-line: Enables batch/shell-directed sessions.

StarTeam SDK

The StarTeam SDK provides the following features and capabilities:

- ◆ Open access to the StarTeam repository for custom solution building or third-party product integration
- ◆ Java API for application portability
- ◆ COM wrapper to support scripting languages through a COM interface layer
- ◆ Microsoft .NET Assembly supported by StarTeam COM objects
- ◆ Support for the StarTeamMPX publish/subscribe technology

Related Concepts

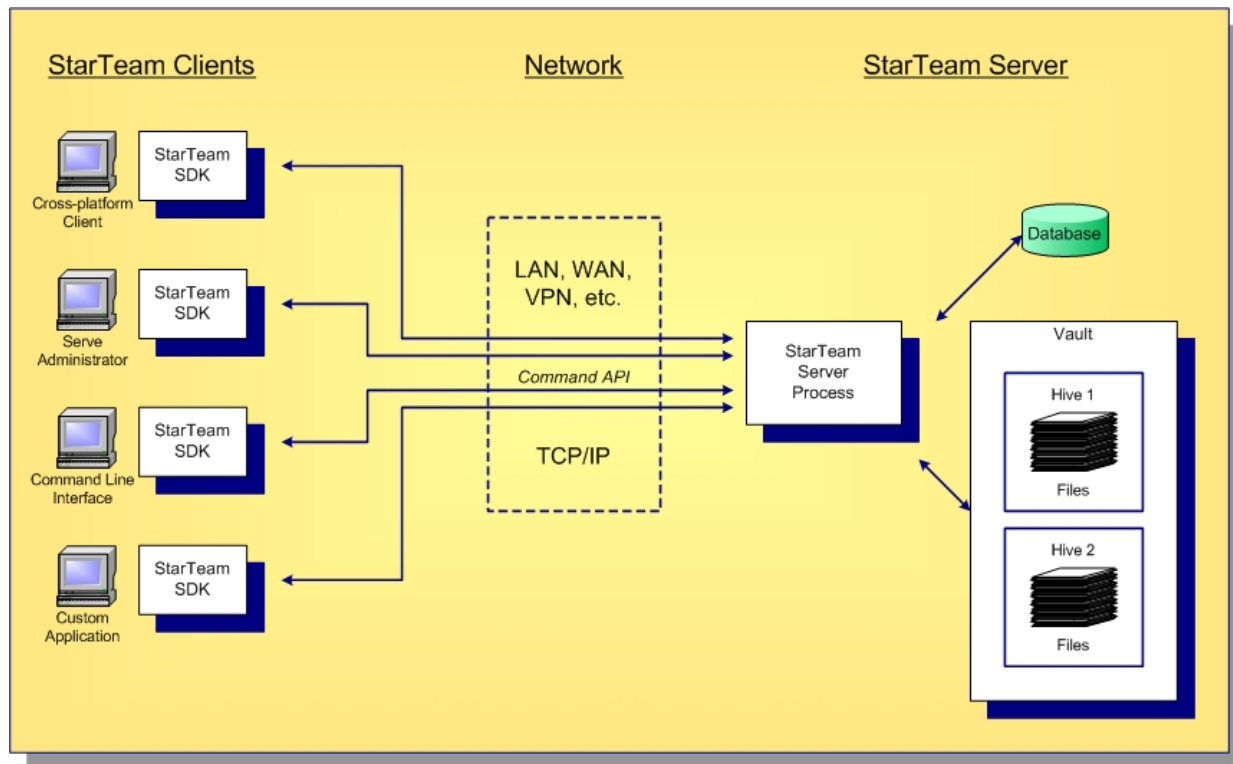
[What's New in StarTeam 2009](#)

[Where to Find Documentation for Each Product](#)

[Tour of the UI](#)

Standard StarTeam Architecture Overview

The standard architecture represents the minimal components present in a StarTeam instance: a StarTeam Server process managing a vault and a database and one or more StarTeam clients. With just these components, all basic StarTeam functionality is available. The core components of the standard StarTeam architecture are depicted below.



StarTeam employs a client/server architecture. The *Cross-Platform Client* (CPC), *Server Administrator* (Server Administration Tool), and *Command Line Interface* are examples of bundled StarTeam clients. StarTeam clients use the freely available *StarTeam SDK*, so you can write custom applications that have access to the same features as the bundled clients. The SDK is fully featured in Java, .NET, and COM flavors, allowing you to write custom applications for any environment. A single StarTeam client can have multiple sessions to any number of StarTeam servers.

All StarTeam clients connect to a StarTeam Server process using TCP/IP, so virtually any kind of network can be used: LAN, WAN, VPN, or the public Internet. StarTeam uses a proprietary protocol called the *command API*, which supports compression and multiple levels of encryption. The command API has been optimized to support high performance, automatic reconnect, delta check-out for slow connections, and other important features.

A single deployment instance of StarTeam is known as a *server configuration*, usually shortened to just *configuration*. The persistent data of a configuration consists of a *database* and a *vault* and is managed by a single *StarTeam Server process*. The database holds all metadata and non-file artifacts, whereas file contents are stored in the vault. The database can be Microsoft SQL Server Express (SSE), full SQL Server, or Oracle, and it can reside on the same machine as the StarTeam Server process or a separate machine. The StarTeam database and vault can be backed-up dynamically, while the server is in use. This supports 24 x 7 operations that want to minimize down time.

StarTeam's vault is a critical component that affects performance and scalability. In contrast to the traditional *delta storage* technique, StarTeam's vault uses an innovative (patent pending) architecture designed for scalability, performance, high availability, and dynamic expandability. Today, customers are storing up to a terabyte of data in a single StarTeam vault, but it was designed to store content up to a petabyte and beyond.

Within the vault, files are stored in containers known as *hives*. A hive is a folder tree containing *archive* and *cache* files on a single disk volume. Hives can be dynamically added on existing or new disk volumes, thereby allowing

virtually unlimited capacity. StarTeam stores each file revision in a separate archive file in a manner that minimizes space usage as well as duplicate content. Amazingly, StarTeam's vault uses less space than delta-based storage. In certain cases where it is more economical to send file deltas to clients instead of full versions, StarTeam generates and caches delta files. However, in most cases sending full versions is more economical.

What's New in StarTeam 2009

This section provides an overview of the new features found in StarTeam 2009.

In This Section

[New Features in StarTeam 2009 Server](#)

New features and changes found in StarTeam 2009 Server .

[New Features in the StarTeam 2009 Cross-Platform Client](#)

Describes new features in the StarTeam 2009 Cross-Platform Client.

[New Features in View Compare/Merge](#)

Describes changes and new features in View Compare/Merge and the VCMUtility.

[Borland StarTeam 2009 Web Client](#)

Describes the new StarTeam 2009 Web Client.

[New Features in Other StarTeam 2009 Components and Products](#)

New features in other StarTeam products for this release.

New Features in StarTeam 2009 Server

This release of the StarTeam 2009 Server includes the following new features:

Online Purge

StarTeam 2009 Server introduces Online Purge.

- ◆ Online Purge allows you to purge data while the Server is running, significantly reducing maintenance downtime.
- ◆ The Online Purge process can be started and stopped using a new Online Purge view in the Server Administration Tool. You can also write an SDK script to control and automate the Online Purge process, which enables you to schedule the purge to start and stop at specific times and avoid purging data during peak usage times.
- ◆ The new Online Purge is faster than the previous offline implementation. Offline Purge is still available in StarTeam 2009, but will be removed in subsequent releases.
- ◆ The changes to Online Purge are being done in phases over a series of releases. In the StarTeam 2009 phase, newly deleted data will be available to purge only after a Server restart.
- ◆ Online Purge is an interactive process which can be stopped and restarted anytime when the server is running. Online Purge records its current execution state and provides the ability to restart from the exact point where it stopped. After a server start, Online Purge has to be restarted manually.
- ◆ You can start and stop Online Purge on a remote Server as well as a local Server.

StarTeam Connection Control

StarTeam 2009 Server allows administrators to fine tune the set of client applications that can connect to the server by customizing a new `app-control.xml` file. This feature prevents unwanted SDK applications from connecting to the Server and draining Server resources.

Note: This is strictly an administrative tool, not a security measure.

app-control.xml Configuration File

The server looks for a new configuration file named **app-control.xml** located in the `AppControl` subdirectory under the StarTeam repository root directory. When a new configuration is created, StarTeam 2009 Server creates this file from a template `app-control.xml` file located in `AppControl` directory under the Server installation directory.

The configuration `app-control.xml` file, if present, contains a set of rules. Each rule asks the server to test the incoming client connections to satisfy one or more of the following conditions:

- ◆ The StarTeam SDK is greater or equal to a certain version.
- ◆ The application name, connecting user name, and/or client workstation name must match a specified text pattern.

The Server tests each incoming client connection against all the rules present in the `app-control.xml` file until a match is found or until the rule list is depleted. Once a match is found, no more checks are done and the connection handshake sequence is resumed. If no match is found, the connection is refused. If the `app-control.xml` file does not exist in the `AppControl` directory, the Server allows all supported client applications to connect.

AllowedApp

AllowedApp: This is the main rule element. It must have a **Name** attribute that specifies the text pattern for the client application name (such as "client identification string"). The text pattern can have an asterisk character (*) that is used as a wildcard. Besides the **Name** attribute, this node can optionally specify one or more of the following attributes:

- ◆ **MinimumSDKVersion:** specifies a minimum version of StarTeam SDK with which the client application is built. The format of this field is **nn.nn.nn.nn**, where **nn** is a non-negative number. Not all of the "dot" numbers have to be specified, for example **MinimumSDKVersion="10.4"** will allow **10.4.x.y** and above (**10.5**, **11.0**, and so on).
- ◆ **WorkStationID:** if set, specifies text pattern to match the client computer name.
- ◆ **Name:** if set, specifies text pattern to match the StarTeam user name.

If an optional parameter is not set, the server does not test the corresponding connection attribute.

AppDefault

AppDefault: This is an optional element that can be used to specify default values for one of the parameters listed under **AllowedApp**. The syntax of this element is similar to the **AllowedApp** syntax, except that the **Name** attribute cannot have a default value. Default values can be specified for **MinimumSDKVersion**, **WorkStationID**, and **UserName**.

Other StarTeam 2009 Server Features

This release of the server includes the additional new features:

- ◆ StarTeam Server for Windows platform is now supported on 64-bit architecture, increasing access to more available memory. This requires Windows Server 2008 64-bit.
- ◆ StarTeam Server 2009 supports all the other new StarTeam 2009 features as well, such as Change Packages and trace support for artifact to artifact linking (external links) across different Servers.
- ◆ StarTeam Server 2009 now creates new projects with only the "File" type pre-selected as a default for new views. Users can still change the project properties after the project is created, and they can change the item types included for any given new view. However, if the user changes nothing, by default new views will only include files when they are created. **Note:** This change does not affect any existing projects. It only affects new projects created with new StarTeam Server 2009 Servers or existing servers once they are upgraded to StarTeam Server 2009. Adding other item types to the Project Properties (after the view is created) will NOT populate the items that were contained in the parent view (but left out during New View creation). If the user wants to bring the previous items into the new view, they must retrieve them by Rebasing from the parent view.
- ◆ StarTeam Server 2009 has improved command handling performance achieved by using Asynchronous I/O to perform network read and write operations on supported Windows 32-bit and 64-bit platforms. This is the second phase of Asynchronous I/O support. The first phase was introduced in StarTeam Server 2008 R2 release and provided only write operation support.
- ◆ A new StarTeam Web Server is being introduced to support the new Web Client being released for the first time in StarTeam 2009.
- ◆ Additional changes have been made to increase and improve Server performance.
- ◆ The Linux Server installation instructions have been moved into the main "StarTeam 2009 Installation Guide" ([ST_Install_en.pdf](#)) at <http://techpubs.borland.com/starteam/>.
- ◆ The StarTeam Server Administration Tool uses the new Eclipse Info Center Help. See "What's New in Documentation" in "New Features in Other StarTeam 2009 Components and Products."

New Features in the StarTeam 2009 Cross-Platform Client

This topic describes the new features and changes in this release of the StarTeam 2009Cross-Platform Client.

The following new features in this release are described in more detail in the sections below:


- ◆ Change Packages
- ◆ External linking
- ◆ EOL Improvements
- ◆ Other Cross-Platform Changes

Change Packages

StarTeam has historically provided many features that support change management (CM), including built-in workflow, customizable workflow, process links, process tasks, and View Compare/Merge (VCM). Now, StarTeam 2009 adds a comprehensive *change package* feature which allows you to track all changes made in a single commit using a change package object. As a result of this new feature, VCM now uses change packages instead of VCM process tasks as it has in the past. Change packages are a change management feature that improves StarTeam ability to manage and track updates. Change packages are an evolution of the View Compare/Merge (VCM) feature first introduced in the StarTeam 2006 release.

For more information on the new Change Packages feature see the topic “What’s New in View Compare/Merge”.

External Linking

A new external linking feature provides the ability to link between items on different Servers (item to item linking across servers). The process for creating external links is basically the same as for creating links between items on the same Server. However, an external link has a decoration () that clearly shows it is an external link rather than a regular link, and there is a new check box on the Link tab that enables you to show or hide external links in the display.

- ◆ All the same linking operations are available for external links that are available for standard links, such as Create Link, Complete Link, and drag and drop operations. To create external links, the projects on both Servers containing the items you want to link need to be opened in the Cross-Platform Client at the same time.
- ◆ The Link pane also gives you the ability to search for external links in all Servers to which you are logged on.
- ◆ When you create an external link, the item details for the external link on the Link pane are represented by a URL so you know how to find that item
- ◆ External link options are the same as standard links, such as being able to view the link properties. However, floating and pinning external links are not available.
- ◆ External links have direction, so whether you are viewing the external link from the source Item of the link, such as a CR, or the Item on the external Server, such as the file being linked to from the CR, the source and target information will always remain the same and does not change based on the item you have selected.
- ◆ You can create external links only to objects that exist on a StarTeam 2009 Server which supports external linking, and you must use the StarTeam 2009 Cross-Platform Client. Only the Source Item of the external link must be on a StarTeam 2009 Server.
- ◆ You cannot create external links to or between Change Packages.
- ◆ Some information which is available for standard links is not available for external links because not all of the information from the external object is available, such as file status, who has the file locked on the external Server, what the object's folder path is, and the folder in which the object actually exists.

- ◆ **Access Rights** dialog boxes now contain **External Links** container level access rights.

EOL Improvements

EOL improvements result in better support for fixed EOL conversion files. For example, EOL conversion now works correctly for unicode files which previously could be corrupted on check-in.

Files can be checked out in LF format on every platform, regardless of specific options. Also, Update Status works for all text files once EOL Format is defined, regardless of what EOL format was used when they were checked-out.

For compatibility with older Clients, if check-out "EOL conversion" is not requested, and EOL Format is Undefined, files are still checked out with the EOL conversion with which they were added to the Server.

StarTeam 2009 provides the following new EOL handling.

- ◆ The property is still displayed as "EOL Character" in the Cross-Platform Client Item pane.
- ◆ The EOL Format property is only meaningful for text files during the check-out operation.
- ◆ The EOL Property values are:
 - *Undefined* (null in the SDK): Used for files added before StarTeam 2009.
 - *Client Defined*: Causes workstation default or per-checkout EOL conversion option to be used.
 - *Fixed CR*, *Fixed LF*, and *Fixed CRLF*: Causes this EOL format to be used always; the work station/check-out conversion option is ignored.
- ◆ The EOL Format property can be set in the Cross-Platform Client in the **Add/Check-in** and **File Properties** dialog boxes.
- ◆ The Cross-Platform Client EOL conversion for the add/check-in options have been removed.
- ◆ By default, the SDK will compute the EOL Format under the following conditions:
 - When a new text file is added or a new revision is checked in for a text file whose EOL Format is *Undefined*, the file's EOL convention matches the platform default, EOL Format is set to *Client Defined*. Otherwise, EOL Format is set to the convention found: *Fixed LF*, *Fixed CR*, or *Fixed CRLF*.
 - The user can change EOL Format to any value (other than *Undefined*) at any time.
 - Regardless of their EOL Format setting, text files added or checked in with a StarTeam 2009 Cross-Platform Client always use a canonical (CRLF) format in the vault.

Note: The default for automatic EOL conversion for check-out operations has been changed to "checked" if the user does not have that option defined already. Users that upgrade to 2009 should check that option to be sure they have it set correctly given the new EOL Format changes.

Other Cross-Platform Client New Features

The following are additional new features added to the Cross-Platform Client:

- ◆ StarTeam 2009 now has "Me" queries that allows a query to be set up which is evaluated against the currently logged in user ("Me"), rather than having to specify a specific username at the time of query creation.
- ◆ The order of the **Folder** context menu has been reorganized to be more like the **Items Table** context menu to improve menu consistency.

- ◆ The **Select View** dialog box has a new checkbox which allows the user to choose to open the selected view in a new View window instead of changing the current view window to the selected view.
- ◆ In the Folder Tree, folder icons have a new decorator to signify that a folder is using an alternate path for the working folder rather than the default working path.
- ◆ When the user creates a new project, the Server previously pre-selected all item types for the project properties. This means that if the user did not change anything in the **Project Properties** dialog box, or in the **New View Wizard**, then new views would contain all item types. As a best practice recommendation, we want to discourage users from including item types other than **File** for new views. StarTeam Server 2009 will now create new projects with only the **File** type pre-selected as a default for new views. Users can still change the project properties after the project is created, and they can change the item types included for any given new view. However, if the user changes nothing, by default new views will only include files when they are created. This change does not affect any existing projects. It only affects new projects created with new StarTeam Server 2009 or existing Servers once they are upgraded to StarTeam Server 2009. **Note:** Adding other item types to the project properties (after the view is created) will NOT populate the items that were contained in the parent view (but left out during new view creation). If the user wants to bring the previous items into the new view, they must retrieve them by using View Compare/Merge to rebase them from the parent view.
- ◆ The Cross-Platform Client now supports comparing properties of non-file items using the embedded compare pane. For example, you can select two CRs in the Item pane, or two historical revisions of the same CR, and choose **Tools ▸ Compare** to compare their properties. The embedded compare window will appear at the bottom of the window displaying the properties of each selected CR. This new feature affects the all **Item** and **Information** tabs in the both the **Content Perspective** and the **Change Perspective**.
- ◆ The **File Check In** dialog box no longer displays frozen revision labels.

New Features in View Compare/Merge

This topic describes the changes and new features in View Compare/Merge and in the [VCMUtility](#).

Change Packages

StarTeam historically provided many features that supported change management (CM), including built-in workflow, customizable workflow, process links, process tasks, and View Compare/Merge (VCM). Now, StarTeam 2009 introduces a comprehensive *Change Package* object which allows you to track all changes made in a single commit. As a result of this new feature, VCM now creates change packages instead of VCM process tasks as it has in the past.

Change packages improve StarTeam's ability to manage and track updates. Change packages are an evolution of the View Compare/Merge (VCM) feature first introduced in the StarTeam 2006 release. A change package is an object that contains a set changes applied to a target view. To create a change package, a user first starts a VCM session, which acts as a staging area where changes are defined, reviewed, and tested.

A VCM session is visible in the target view as a change package after it has been saved or committed.

- ◆ As a result of using the new change package item, merge points and process tasks are no longer used in VCM sessions to track changes.
- ◆ Perspectives have been introduced into the StarTeam View window so you can click an icon to switch between the standard **Content Perspective**, represented by the StarTeam logo icon, and the **Change Perspective**, represented by a new icon next to the standard icon. These new perspective icons are right-justified in the toolbar. Using the **Change Perspective**, a manager, for example, can review all the change package objects and the details around those change packages, see what changes were committed to a view, and review changes in uncommitted change packages being proposed for committal.
- ◆ VCM Session menu items have changed. For example, now that the change package is an object, a new **Change Package ▸ Properties** menu item exists for opening the change package to view and/or change its properties, such as the working folder for the VCM session. Other menu changes are the **Change Package ▸ Save** menu item that saves a change package object in an uncommitted state onto the target view of the StarTeam Server. **Change Package ▸ Export** is still supported as the way to save a VCM Session as a `.vcmx` file to send to others for review, though change packages are now preferred over `.vcmx` files. For additional information on the new **Change Package** menu, see the "Change Perspective UI" topic in "Getting Started" under "Tour of the UI".
- ◆ Since each change package is a new object in StarTeam that represents a set of changes, StarTeam 2009 has added a new **Change** tab at the bottom of the Client to support viewing updates to a selected item that occurred as the result of a change package. A more detailed explanation of all the parts of the **Change Perspective** is available in the "Tour of the UI" section of "Getting Started" in the "StarTeam Help".
- ◆ Other options are available in the **Change Perspective** such as toolbar buttons, a standard **Filter** menu, and **Change Layout** toolbar button which lets you switch between a left/right layout or a top/bottom layout. In the left/right layout, icons are used for switching viewers in the **Change Perspective** between the **Detail**, **History**, **Label**, **Change**, and **Replay** panes. In the top/bottom layout, bottom tabs allow you to change these views.
- ◆ A context menu item now lets you copy URLs to the clipboard to saved change packages so others can open the URL to the target view and review the change package in read-only mode, eliminating the need to send a `.vcmx` file of the whole VCM session.
- ◆ In the **Replay** viewer, you can use drag and drop to replay a change package to another view. You can also re-open the change package in a VCM session using the item's **Advanced ▸ View Compare/Merge** menu option to initiate a **Replay** to another view.
- ◆ You can delete a change package if it is uncommitted. Once it is committed, the delete operation is no longer available.

- ◆ You can create exclusive locks on uncommitted change packages. Only one person can make changes to a change package at a time. Consequently, a change package is automatically locked exclusively when it is opened for editing.
- ◆ A **Restart Session** menu operation in the **Change Perspective** allows a user to restart (re-compare) a change package in a VCM session.
- ◆ The VCM Session menu now has two new items: **Copy URL to Clipboard** and **Select in View**. When the current VCM Session has been saved as a change package, these operations allow the user to more easily access saved change packages.
- ◆ A new change package **Report** menu is available from the **Reports** dialog box for change packages.
- ◆ The **Change Perspective** supports the **Compare Properties** menu for comparing change package properties.

Other VCM Changes

Other new features in View Compare/Merge, besides the already mentioned Change Packages and External Linking, are the following miscellaneous changes:

- ◆ The **View Compare/Merge Wizard** has a new **Exclude Properties** page which allows you to exclude properties of any branchable item type. It lets you select specific item type properties for which you do not want to merge changes when the session is committed.
- ◆ The **View Compare/Merge Wizard** has a new button at the bottom right of each wizard page which, when clicked, opens an information pane that shows you the details of the choices you have made for the VCM session up to that step in the **View Compare/Merge Wizard**. To hide this information pane, click the button again and it will disappear. The advantage of this information pane is that you can review in one place all the details of the session before you click **Finish** without having to go back and forth to different pages of the wizard to see what you selected. That way, if you decide you want to make a different choice, you can go back to the specific page of the wizard, make the change, then click **Finish** at that point.
- ◆ The **Compare**, **Merge**, and **Test** perspective buttons have been moved to the right side of the toolbar.

Merge Points

Merge points are no longer used in VCM sessions now that the change packages have been implemented. VCM now uses internal "change" objects that yield improved merge capabilities compared to merge points.

New or Changed Difference Types

The following changes to difference types and actions have been made:

- ◆ *Fail* has been changed to *Needs Review*. *Needs Review* is now a legal action in all cases.
- ◆ Custom merge types: Changing the default action of any difference type to *Needs Review* means that human intervention is required before a commit can be made.
- ◆ *Mark Resolved* no longer creates a Merge Point. Now it means create a *Mark Resolved* Change Object. *Mark Resolved* is now a legal action in many cases where it was not previously permitted. This affects about two dozen difference types.
- ◆ *Don't Care* difference types have been fixed. We now distinguish between the *Modified in target* and *Unmodified in target* cases.

For example, in StarTeam 2008 Release 2,

[2000]: Moved in source, target on different branch

```
ItemPresentInSource: true.
ItemPresentInTarget: true.
ItemDeletedInSource: false.
ItemDeletedInTarget: false.
ItemModifiedInSource: false.
ItemModifiedInTarget: Don't Care.
ItemMovedInSource: true.
ItemMovedInTarget: false.
ItemsInDifferentBranches: true
Default Action: Ignore.
Legal Actions: Ignore; Move; Move and Overwrite
```

In StarTeam 2009,

[2000]: Moved in source, target on different branch

```
ItemPresentInSource: true.
ItemPresentInTarget: true.
ItemDeletedInSource: false.
ItemDeletedInTarget: false.
ItemModifiedInSource: false.
ItemModifiedInTarget: false.
ItemMovedInSource: true.
ItemMovedInTarget: false.
ItemsInDifferentBranches: true
Default Action: Ignore.
Legal Actions: Ignore; Move; Needs Review; Mark Resolved
```

Old rows which now have **Modified in target=false**

```
[2000]: Moved in source, target on different branch
[2500]: Moved and modified in source, target on different branch
[2510]: Moved and modified in source, target on different branch, same content.
```

New rows with **Modified in target=true**

```
[2060]: Moved in source, branched and modified in target
[2520]: Moved and modified in source, target on different branch, modified in target
[2530]: Moved and modified in source, target on different branch, modified in target,same
content
```

Changes to the VCM Table of Action Decisions

The following changes have been made to the VCM Table of Action Decisions for StarTeam 2009:

```
[100]: Parent folder failed

ParentFolderFailed: true

Default Action: Needs Review
Legal Actions: Ignore; Needs Review

[110]: Parent folder ignored
```

```

ParentFolderIgnored: true

Default Action: Ignore
Legal Actions: Ignore; Needs Review

[200]: Target folder has floating share in source view

TargetFolderHasFloatingShares: true

Default Action: Needs Review
Legal Actions: Ignore; Needs Review

[620]: Deleted in target (Promote)

MergeType: Promote
ItemPresentInSource: true
ItemPresentInTarget: false
ItemDeletedInSource: false
ItemDeletedInTarget: true

Default Action: Ignore
Legal Actions: Ignore; Share; Reverse Share; Needs Review

[600]: Deleted in target

ItemPresentInSource: true
ItemPresentInTarget: false
ItemDeletedInSource: false
ItemDeletedInTarget: true

Default Action: Ignore
Legal Actions: Ignore; Share; Needs Review

[520]: New in source, shared (Promote)

MergeType: Promote
ItemPresentInSource: true
ItemPresentInTarget: false
ItemDeletedInSource: false
SourceItemOnRootBranch: false

Default Action: Needs Review
Legal Actions: Ignore; Share; Needs Review

[510]: New in source (Promote)

```

Changes in Resolving Process Tasks

Resolving a Process Task in 2008 Release 2,

- ◆ You had to follow the process links.
- ◆ Process links could not reference a deleted item
- ◆ You could not propagate *deletes* using process item scope.

Resolving a Process Task in 2009 involves the following:

- ◆ Opening the attached *.vcmx file.

- ◆ Using the *ItemDifferences* to determine scope

This is the equivalent of using change package/change objects.

VCMUtility Command-line Changes

The `VCMUTILITY` is integrated with change package objects. New commands and session options have been made to support change packages.

The following additions have been made to the `VCMUTILITY` command to support change packages:

Command Options

- ◆ {Open <Change Package name>}
- ◆ {Replay <Change Package name>}

Session Options

- ◆ {Description <description>}
- ◆ {Name <Change Package name>}

Other Syntax Options

- ◆ <Change Package name>
- ◆ <folder path>
- ◆ <VCM exchange file>
- ◆ <VCM session file>

These, and all the other `VCMUtility` commands and options are listed in the Compare/Merge Reference section of the Cross-Platform Client Help.

Related Concepts

[Change Packages Overview](#)

Borland StarTeam 2009 Web Client

The new Borland® StarTeam® Web Client is an intuitive Web-based interface that multiple simultaneous users can use to connect to one or more StarTeam Servers to access projects and manage items.

This initial release of the Web Client delivers a core feature set designed to meet the needs of users responsible for viewing, creating, and editing StarTeam change requests, requirements, tasks, and topics.

Web Client Capabilities

The StarTeam Web Client supports the following activities:

- ◆ Using public filters on the StarTeam server to refine the scope of items to browse
- ◆ Creating a non-file Item
- ◆ Editing item properties
- ◆ Locking and unlocking an Item
- ◆ Displaying item details
- ◆ Deleting an item
- ◆ Downloading a file to a local or network drive
- ◆ Starting a view session with a generated item or folder URL
- ◆ Viewing an Item's historical revisions
- ◆ Viewing a revision's properties
- ◆ Editing a revision's comment

Note: You must possess a StarTeam user license to use the Web Client.

New Features in Other StarTeam 2009 Components and Products

The following are new features or improvements made in other StarTeam products included with this release.

- ◆ What's New in Documentation
- ◆ What's New in StarTeamMPX
- ◆ What's New in Layout Designer

What's New in Documentation

For StarTeam 2009, in our basic applications we have changed from the proprietary Borland Help Browser to the Eclipse Info Center for our online help presentation. Ultimately the Eclipse Info Center will be used across all Borland products.

The Eclipse Browser will be introduced in this release in the Cross-Platform Client, the Server Administration Tool, and the Layout Designer. The combined help documentation called "Administering and Using StarTeam" in previous releases has been replaced with smaller pieces of documentation which are relevant to the application being used. So, for example, the Cross-Platform Client will contain the Client Help, and Help on the Command-line tools, including the [VCMUtility](#), and the Server Administration Tool will contain only the Server Administration Help, plus the help for Command-line tools.

The advantages to you from the Eclipse Info Center is that you can do full-text search, and you can print small sections or whole sections from the Table of contents.

As always, the Help is also available from the Start menu on Windows. On Linux or Solaris, it will be in / [PRODUCT_NAME/Documentation](#) folder.

Note: The Linux Server installation instructions have been moved into the main Installation Guide.

What's New in StarTeamMPX

StarTeamMPX 2009 has the following new features:

- ◆ The Multicast option has been removed from StarteamMPX.
- ◆ Clients subscribe to a new [STEvent3](#) stream which uses more granular subjects for view-specific events. Messages are compressed and batched by transaction. Each Client receives 70% to 80% reduction in traffic. A StarTeam 2009 Client can get as little as 2% of the messages and 2% of the traffic that a pre-StarTeam 2009 Client gets.
- ◆ Certain "duplicate" cache messages are eliminated, for example, redundant file content messages. This reduces traffic to Cache Agents.
- ◆ The message improvements are transparent to Clients.
- ◆ When pre-StarTeam 2006 Clients connections are not allowed, which occurs when the Server minimum API level is > 1.25, the [STEvent](#) event stream is not broadcast. Similarly, the [STEvent2](#) event stream is not broadcast when the Server minimum API level is > 1.66, which means that only StarTeam 2009 and later clients are allowed.
- ◆ New StarTeam 2009 events are sent to the [STEvent3](#) stream, such as change packages and trace objects (external links).

What's New in the Layout Designer

The following features are new for Layout Designer in StarTeam 2009

- ◆ The forms provided in the Cross-Platform Client are now available as example Layout Designer forms.
- ◆ The Layout Designer uses the new Eclipse Info Center Help. See “What's New in Documentation”.

Help on Help

This section describes the Help system for StarTeam. It also explains where to find documentation for each of the StarTeam products.

In This Section

[StarTeam Overview](#)

This topic describes the Help system for StarTeam.

[Where to Find Documentation for Each Product](#)

This topic describes the various methods for accessing the StarTeam product documentation and provides a list of what documentation ships with each of the StarTeam products.

[User Roles and StarTeam Documentation](#)

This topic contains information about various user roles and how the StarTeam documentation ties to those roles.

StarTeam Overview

The StarTeam Help system contains conceptual topics, procedural how-to's, and reference information, allowing you to navigate from general to more specific information as needed.

Conceptual topics	The conceptual overviews provide information about product architecture, components, and best practices for working with StarTeam. At the end of most of the topics, you will find links to related, more detailed information and/or procedural or reference topics.
Procedure topics	The how-to procedures provide step-by-step instructions. For operations in StarTeam that include several subtasks, there are core procedures, which include the subtasks required to accomplish a larger task. If you are beginning a task, such as upgrading a server (in the installation guide), and want to know what steps are involved, see the core procedure for the area you are working on. In addition to the core procedures, there are several single-task procedures. All of the procedures are located under the Procedures area of the consolidated help system. Additionally, most of the conceptual and reference topics provide links to the pertinent, related procedures.
Reference information	The reference topics provide detailed information on subjects such as command line options, StarTeam fields, and file status information. All of the reference topics are located under the Reference area of the consolidated help system, and most of the reference topics provide links to related procedural or conceptual topics.

The StarTeam Help system has four main areas: Getting Started, Concepts, Procedures, and Reference. Each of the main areas contain subareas that group information into functional areas as described in the table below.

This help area...	Contains information about...
General Operation	Procedures and conceptual information for a developer or occasional user of StarTeam, such as checking files in and out and setting personal user options.
Customization	Procedures and conceptual information for a user that customizes StarTeam with the Layout Designer.
Project Administration	Procedures and conceptual information for a StarTeam Project Administrator, such as creating projects and views.
Server Administration	Procedures and conceptual information for a StarTeam Server Administrator, such as customizing server configurations, backing up information and migrating servers.
Security	Procedures and conceptual information for a StarTeam Server Administrator interested in security features available for StarTeam, such as managing users, groups, access rights, and passwords.
Configuration	Procedures available to configure the StarTeam clients, such as adding a server configuration and changing a password.
Reporting and Testing	Procedures available in the StarTeam clients for a QA Engineer or Project Manager, such as creating charts or reports, working with change requests, and querying or filtering data.

Related Concepts

[StarTeam Product Overview](#)

[Where to Find Documentation for Each Product](#)

Where to Find Documentation for Each Product

This topic describes the various methods for accessing the StarTeam product documentation and provides a list of what documentation ships with each of the StarTeam products.

How to Access Product Documentation

In general, you can access the documentation for the StarTeam products as follows:

- ◆ From the **Help** menu within the product.
- ◆ If using a Windows system, you can locate documentation for the StarTeam products by accessing the **Start** ► **Programs** ► **Borland StarTeam** ► **<Product>** ► **Documentation** menu. The **Documentation** menu lists all of the available documentation for the selected product.
- ◆ Readme files and installation instructions can be found directly under the root installation directory (or on the root of the installation CD). For documentation available in other languages (Japanese, French, or German), the language-specific versions of the release notes and installation instructions are indicated with and appropriate *_countrycode* in the filename. For example, `readme_ja.html` contains release note information for the Japanese language. PDF manuals are located in the Documentation subfolder on the product CDs.
- ◆ PDF manuals and online help files can be found in the PDF and Help subfolders in the root installation folder.
- ◆ You can download documentation directly from the Borland StarTeam Technical publications web site: <http://info.borland.com/techpubs/starteam>.

StarTeam Product Documentation

Certain portions (but not all) of the StarTeam documentation set have been consolidated into one help system for this release. Each product and its associated documentation follows.

StarTeam Server and StarTeam Cross-Platform client Documentation:

This documentation is available in English, Japanese, French, and German languages.

StarTeam Help (Online help)	Online help version of StarTeam Help that opens from the Help menu within the Server Administration Window, Cross-Platform Client, and the StarTeam Visual Studio 2005 Integration.
Administering and Using StarTeam (AdministeringAndUsingStarTeam.pdf)	An identical version of StarTeam Help available in PDF format.
StarTeam Extensions User's Guide (extensions.pdf)	A PDF version of help for StarTeam Extensions.
StarTeamMPX Administrator's Guide (adminMPX.pdf)	The PDF version of help for StarTeamMPX.
Install_en.pdf	The language-specific version of the StarTeam installation guide covering many of the products in the StarTeam product line.
readme_en.html	The language-specific version of release notes covering many of the products in the StarTeam product line.

StarTeam Web Client

This documentation is available in English, Japanese, French, and German languages.

Web Client Help (Online Help)	Eclipse Browser help for the Web Client Help opens from within Web Client on the Help menu.
-------------------------------	---

StarTeamMPX

This documentation is available in English, Japanese, French, and German languages.

StarTeamMPX Administrator's Guide (adminMPX.pdf)	The PDF-version of the administrator guide for this product.
Installing StarTeam (Install_en.pdf)	The language-specific version of the StarTeam installation guide covering many of the products in the StarTeam product line.
Release Notes (readme_en.html)	The language-specific version of release notes covering many of the products in the StarTeam product line.

StarTeam Workflow Extensions

This documentation is available in English, Japanese, French, and German languages.

StarTeam Extensions User's Guide (extensions.pdf)	The PDF-version of the user guide for this product.
Installing StarTeam (Install_en.pdf)	The language-specific version of the StarTeam installation guide covering many of the products in the StarTeam product line.
Release Notes (readme_en.html)	The language-specific version of release notes covering many of the products in the StarTeam product line.

Borland LDAP QuickStart Manager

This documentation is available in English, Japanese, French, and German languages.

LDAP QuickStart Manager Guide (LDAPQuickStart.pdf)	The PDF-version of the user guide for this product.
Release Notes (readme_LDAP_en.html)	Release notes covering LDAP QuickStart Manager.
Installation Instructions (install_LDAP_en.html)	Installation instructions for LDAP QuickStart Manager.

Borland Search

This documentation is available in English, Japanese, French, and German languages.

Borland Search Administrator's Guide (SearchInstallAdmin.pdf)	The PDF-version of the user guide for this product.
Release Notes (readme_BorlSearch.html)	Release notes covering Borland Search.
Installation Instructions (install_BorlSearch.html)	Installation instructions for Borland Search.

StarTeam SDK

SDK Programmer's Guide	HTML-version of the programmer's guide for the StarTeam SDK.
Java API Reference	Java doc for the StarTeam SDK.
COM API Reference	COM building blocks for the StarTeam SDK.
Release Notes (readme_SDK.html)	Release notes covering the StarTeam SDK.

StarTeam Datamart

StarTeam Datamart User's Guide (StarTeam Datamart User Guide.pdf)	The PDF-version of the user's guide for this product.
Release Notes (readme_Datamart_en.html)	Release notes covering StarTeam Datamart.

StarTeam Import/Export Manager

StarTeam Import/Export Manager User's Guide (stiemgr.pdf)	The PDF-version of the user guide for this product.
Release notes (readme_IEM.html)	Release notes specific to StarTeam StarTeam Import/Export Manager.
Installation Instructions (install_IEM.html)	Installation instructions for StarTeam StarTeam Import/Export Manager.

StarTeam Toolbar Utility

Using the StarTeam Toolbar (SBToolbar.pdf)	The PDF-version of the user guide for this product accessible by clicking the Help button in the StarTeam Toolbar Utility.
--	---

Related Concepts

[StarTeam Product Overview](#)

[StarTeam Overview](#)

[Tour of the UI](#)

User Roles and StarTeam Documentation

This topic contains information about various user roles and how the documentation ties to those roles. These roles may or may not describe the roles within your organization, but are provided as an example of where operations might be divided among StarTeam users.

The suggested roles for StarTeam are given below with details about where to find associated procedures and conceptual topics related to these roles.

User/Developer	A user primarily concerned with checking files in or out, merging files, and closing change requests. Refer to the General Operation nodes in the consolidated documentation set for concepts and operations associated with this user role.
Project Manager/Super User	This role is made up of procedural and conceptual information from the General Operation, Project Administration, and Server Administration sections.
Tester	A user that works on a QA team would fall under this role. Topics from the General Operation and Tester sections of the documentation would cover information for users in this role.
Administrator	This role deals with installation, configuration, and maintenance for StarTeam. Topics from the Server Administration and Installing and Configuring StarTeam sections in the documentation contain conceptual and procedural information for this user role.
Customizer	A user that would provide customization to StarTeam using alternate property editors, features installed with StarTeam Extensions, and the StarTeam Layout Designer. Within the documentation, topics about the StarTeam Layout Designer and creating custom property fields are located in the Customization sections. You can refer to the StarTeam Extensions User's Guide (extensions.pdf) for information about using alternate property editors and modifying the built-in custom workflow provided with StarTeam.

StarTeam Basics

The topics in this section provide an overview of basic StarTeam concepts.

In This Section

[Containers](#)

Describes the concept of containers as it relates to StarTeam.

[Artifacts](#)

Discusses the concept of artifacts as it relates to StarTeam.

[Artifacts Versus Items](#)

Describes the differences between artifacts and items in StarTeam.

[Folders](#)

Provides an overview of the folder hierarchy.

[Files](#)

Describes the behavior of files under version control.

[Change Requests](#)

Describes the change request component and its built-in workflow.

[Requirements](#)

Describes the requirements component.

[Tasks](#)

Describes the task component.

[Topics](#)

Describes the topic component.

[Links: Internal and External](#)

Discusses the concept of links as it relates to StarTeam.

[Labels](#)

Describes the concept of labels as it relates to StarTeam..

[Branching, Merging and Dot Notation](#)

This topic explores the concepts of branching, merging and dot notation as they relate to StarTeam.

[Sharing and “Cheap Copies”](#)

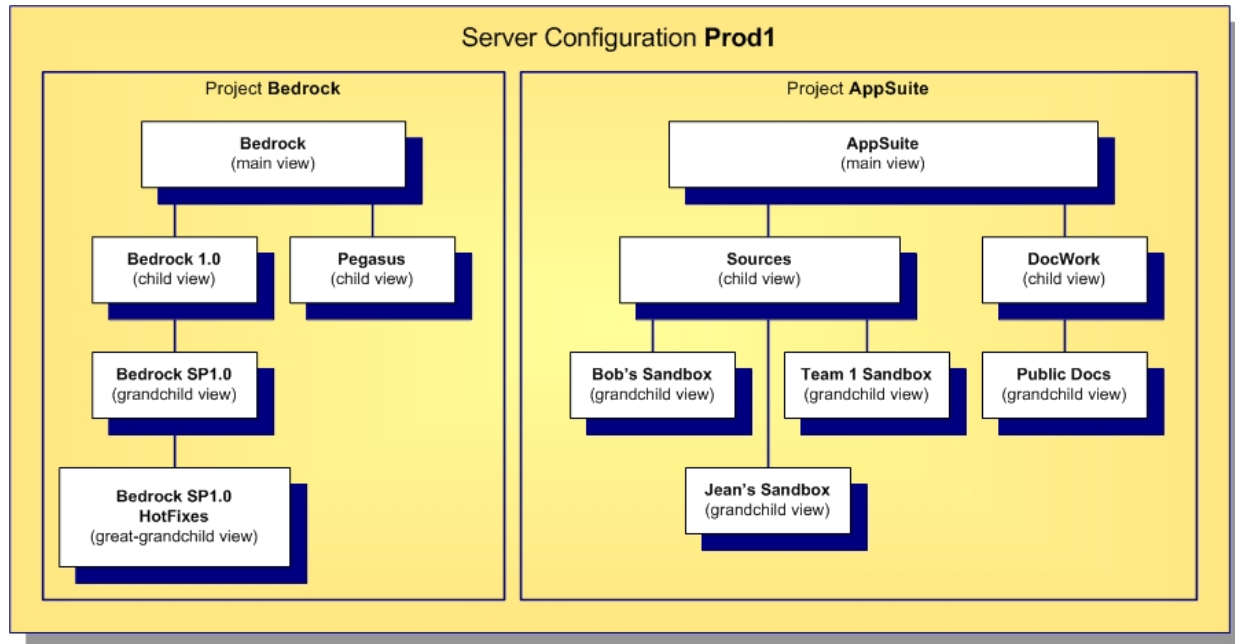
This topic explains the concept of sharing as it relates to StarTeam.

[Promotion States](#)

Describes the concept of promotion state as it relates to StarTeam.

Containers

StarTeam server configurations, projects, and views are *containers* that allow you to organize artifacts based on application, module, business unit, or other criteria. These three basic containers are illustrated in the diagram below:



Server Configurations

A *server configuration* is also referred to as a repository or as an instance. All files, change requests, and other artifacts that can be interrelated and managed as a whole reside in the same configuration.

Note: Throughout the documentation, the terms *server configuration* and *server* are also used interchangeably. This is because each server configuration is often deployed on its own server machine, managed by its own StarTeam server process. However, be aware that StarTeam allows multiple server configurations and server processes on a single machine, so the server configuration-to-machine relationship does not have to be one-to-one.

Projects

Within a server configuration, artifacts are organized into *projects*, which group and manage related items hierarchically in a set of folders. Creating a project allows you to put files under version control, set requirements, track change requests, manage tasks, audit user actions, and discuss the project. Each project has at least one view, called the initial or root view. For example, a project for a software product might include files on the product's functional specifications, marketing requirements, source code, and test suites, all stored in separate folders in the initial view. As the product progresses from one release to another, additional views of these folders can be created. One view could represent the 1.0 version of the product, while a second view represents the 2.0 version, and so on.

Before a server configuration can be used, at least one project must be created. A server configuration can hold multiple projects, each oriented to the lifecycle needs of a specific team, application, or component. The configuration in the diagram above has two projects: **BedRock**, perhaps for foundation components, and **AppSuite**, which could be used for applications belonging to a common suite.

Views

Each StarTeam project consists of one or more *views*. Think of a view as a “subproject”—it is a subset of the project’s contents that support a specific activity. Every project automatically receives a *main view* through which folders, files, and other objects can be organized. Additional *child views* can be created to represent subsets of the main project information, historic snapshots of project information, or work areas for new development work. StarTeam provides a wide range of options for view creation to support a wide range of development scenarios.

Workspaces

In addition to the three basis containers, StarTeam supports a client-side container called a *workspace*. A workspace is a folder hierarchy located on your computer or in your personal directory on a shared file server. However, the project does not have to exactly match your working folder and its child folders. For example, you may omit child folders in the working folder from a project or copy only specific child folders in an existing project to the working folder. When you add or check in files, the application copies the files from the working folder into the repository. When you check files out, the application copies the files from the repository into the working folder.

In addition to providing a well-defined area for check-in/check-out operations, a workspace allows StarTeam to compute the status of each file: which ones have been modified since they were checked-out, which ones are out-of-date, and so forth.

Artifacts

A typical software development lifecycle requires the development, evolution, and management of things other than source files such as requirements, models, graphics, change requests, schedules, tests, and so on. The term *artifact* refers to the generalization of objects that can be versioned, branched, merged, etc. StarTeam supports non-file artifact types directly, providing type-specific behavior for storage, versioning, merging and so forth. All artifacts are versioned, and some are branchable.

The built-in artifact types supported by StarTeam are summarized below:

- ◆ **Folder:** Every view has one *root folder*, which typically has a tree of subfolders beneath it. Folders are patterned from the file system concept of directories. In many cases, you will want to create StarTeam folders that mirror specific directory structures. However, StarTeam folders can hold **any** kind of artifact—not just files. This concept may seem strange at first, but when you discover that you can organize change requests, tasks, and other non-file artifacts the same way you organize files, you will find this feature very powerful. Folders can branch, allowing the same folder to have different properties in each branch.
- ◆ **File:** StarTeam allows you to store any kind of file: text or binary, authored or generated, small or very large. A few more features are provided for text files such as *keyword expansion* and *EOL conversion*, but all file types otherwise are treated identically. StarTeam allows single file revisions larger than 4GB. Files are branchable, allowing parallel version streams and merging.
- ◆ **Change Request:** A change request (CR) is a general artifact that can represent a defect, enhancement request, or another reason for a software change. Because CRs are often the centroid of change management, the CR type is frequently extended with custom fields, custom GUI forms, and workflow rules. CRs can branch, allowing parallel modifications to the same CR for separate activities such as fixing the same defect in multiple releases. Using integration tools, you can import CRs from and keep them synchronized with other defect management systems.
- ◆ **Task:** StarTeam tasks are modeled after project management tasks: they can be arranged hierarchically to represent task decomposition, they can be connected with predecessor/successor relationships, and they can be updated with progress units known as *work records*. You can import tasks from a project management system such as Microsoft Project, update and maintain them via StarTeam, and then synchronize them back to the original project source. In StarTeam, tasks are versioned but they do not branch.
- ◆ **Topic:** A topic is very similar to a newsgroup message. Like newsgroup messages, topics can be organized into conversation threads. Because topics are artifacts, they are versioned (but not branched) and are stored in the repository with other artifacts. This allows you to capture more application lifecycle “knowledge” such as important discussions related to a design decision or a requirement approval.
- ◆ **Requirement:** If you do not have a formal requirements management (RM) tool, StarTeam requirements provide a convenient, lightweight artifact with which requirements can be captured. Requirements can be arranged hierarchically to represent decomposition, and they can be linked to other artifacts. Since requirements are independently-versioned artifacts, they are more accessible than requirements buried in documents, which are versioned at the whole-document level. If you use a requirements management system such as Borland’s CaliberRM, those “formal” requirements can be imported as StarTeam requirements and organized together with other lifecycle artifacts. (Borland provides integration tools to import, synchronize, and even link artifacts between StarTeam and CaliberRM.) Requirements do not branch.
- ◆ **Audit:** An audit is a read-only “change event” artifact that is automatically generated for other artifact changes: add, modify, delete, move, label attach, etc. Because audits are automatically generated and immutable, they are not really artifacts per se, but StarTeam allows you to access them with similar GUI and SDK techniques as other artifacts, so you can think of them as read-only artifacts. The generation of audits and the length of time that they are retained are configurable.

These artifacts are all “bundled” with StarTeam; however you’re not obligated to use them all. The code for each artifact type is encapsulated in a dynamically-loaded plug-in module called a *server-side component* (SSC). Each SSC is a code library suffixed with .ssc that resides in the server’s installation directory. If you rename an .ssc module before the server starts, the corresponding artifact type will not be used. For example, if you want to use StarTeam as a VCS only, just rename all *.ssc modules except for file.ssc. Note that you always get folders, so there is no .ssc

module for it. Also, we recommend you keep audit.ssc due to the value of the “change log” represented by audit artifacts.

Artifacts Versus Items

What is the difference between artifacts and items? Because you can only access and update artifacts through items, and because StarTeam blends item and artifact properties into a single object (both graphically and in the SDK), you may think of them as a single concept. Although either term works equally well in most cases, we usually use the term *item* when we mean to include folder/view context that items add to artifacts. When the context is not important to the discussion, we use the term *artifact*.

The most important things to know about items are summarized below:

- ◆ **Artifacts can only be accessed through items:** With StarTeam, you can only fetch or update an artifact by directing your request to a specific item. There are no commands to directly access an artifact independent of an item. This means that all artifact access is influenced by the context of the associated item such as its parent folder and the view in which it lives.
- ◆ **Items form folder trees:** “Paths” are formed by each view’s “item tree”. This means that folder artifacts do not define their contents. Instead, what appears inside a folder is determined by the items that refer to the folder’s item as their parent. You don’t really move artifacts—you move items. Moving an item from one folder to another causes the item’s parent to be modified—the artifact referred to by the item isn’t touched. Under the hood, items are versioned similarly to artifacts. This means that changes such as moving an item to a new folder really creates a new item revision, causing the previous item to become historic.
- ◆ **Items facilitate sharing:** Items allow an artifact to appear in multiple folders, views, and projects. To make an artifact—including its entire history—appear in a new location, we only have to create a new item, which is pretty cheap. Sharing is analogous to “hard links” used in UNIX file systems.
- ◆ **Items influence version behavior:** An item has properties that control what artifact revision is referenced and how updates through the item are handled. Items store an *OID* that determines what artifact branch is referenced. An item also stores a *configuration timestamp* to indicate whether it *floats* to the tip revision or is *pinned* to a specific revision of the referenced branch. An item’s *branch-on-change* (BOC) flag indicates if the referenced artifact should branch when modified through the item. For example, if an item currently refers to artifact revision 1.7, and BOC is true, and an update is directed at the item, the artifact is branched by storing the updates to a new revision identified as 1.7.1.0. Additionally, the item is modified to point to the new branch (since it has a new OID), and its BOC flag is set to false. Note that BOC cannot be true for items that point to artifacts that can’t branch (such as topics). Also, an item with BOC equal to false and a pinned configuration timestamp is read-only because we can’t update a historic revision and we can’t start a new branch!
- ◆ **Items create promotion trees:** This is an advanced concept, so we’ll just touch on it briefly here. Items that are shared to a new location “remember” the item from which they were shared. This “share parent” relationship is different than the “containment parent” relationship that forms item paths. It facilitates a concept called *automatic promotion*.

Folders

This topic provides an overview of planning a folder hierarchy and provides some tips for working with new and existing folders. The project or server administrator usually creates projects and project views. If you are a typical user, you routinely open a particular project view and manage *your* folders and their contents, such as files and change requests. Managing application folders is very similar to managing a project. You can create folders, delete folders, and modify their properties—if you have the correct access rights.

Folder Hierarchy

When you create projects, you typically select locations on your workstations as the working folders for those projects. The working folder designated for a project also becomes the working folder for the project's root view and for the root folder in that view's folder hierarchy.

StarTeam treats folders as both containers and items. You can group items within a project view by placing them into folders. For example, a folder named *Source Code* can contain source code files and requested changes to those files. You can create folders automatically when you create a project, or add folders after you create the project. Project or server administrators (or team leads – this all depends on your organization) usually create projects, but anyone can create projects if they have the correct access rights. See your server administrator if you have questions regarding the access rights assigned to you.

When you create a project, StarTeam automatically creates the parent or root folder for that project at the same time. It is actually the root folder of the project's root (or initial) view. The project, view, and this root folder initially have the same name (although those names can be changed).

Usually, the user who creates a project sets up a hierarchy of folders on a workstation before creating the project. The user designates the root folder of that hierarchy as the project's working folder. Then the application can automatically create an application folder for each of the child folders in the hierarchy. The child folder becomes the application folder's working folder.

If child application folders are created at the time the project is created, then:

- ◆ The application folders' working folders were part of an existing hierarchy on the project creator's workstation.
- ◆ Their names are the same as the names of their working folders, but they can be changed later.
- ◆ Their working folders remain hierarchically connected to the root folder's working folder. That is, if you change the path to the root folder's working folder, you also change the path to this folder (unless you manually set an absolute path for these working folders). In other words, the application stores a relative path to each child folder.

One of the most important properties to notice about your folder is its working folder. You will need to know where on your workstation the application will copy file revisions that you check out so that you locate those revisions as needed for modifications. A number of other operations can be performed on folders, such as moving a folder or changing its branching behavior.

A working folder is a property of the folder and represents the actual location on your workstation where StarTeam saves files that you check-out. Despite the fact that these are both called folders, the working folder and the folder are not identical. Their differentiating characteristics include:

- ◆ The path to the working folder can be totally different from the path within the application to the application folder.
- ◆ An application folder is an object controlled from within the application. The data associated with this folder is stored in the database that stores all the project data.
- ◆ A working folder is an object controlled by your operating system. It stores files that are checked out from the application.

A project, its root view, and the root folder of the root view all have the same working folder. For additional views, each view and its root folder have the same working folder.

The working folder for the view/root folder always has an absolute path (starting with the drive letter and specifically naming the folders at subsequent levels until you reach the working folder itself).

If you look at the properties for the root folder, you will see that the working folder is the same. However, it is displayed in the **Complete Working Folder Path** display box instead of the **Default** text box. Since you can only change the working folder at the view level, all of the text boxes for the root folder's working folder are always disabled.

For the child folders that were created at the same time as the project, the application stores the path to each working folder as a relative path.

Folders and Views

You can add new folders and Not-in-View (NIV) folders to views in StarTeam. A NIV folder is a folder on your local disk that does not map to a folder in the StarTeam repository. A NIV folder is displayed as a white folder with a black, dotted border. NIV folders (as with NIV files) do not necessarily need to be added to a view, but you may choose to do so if you just created it and want it to be part of the StarTeam view. However, if the folder is NIV because someone else deleted it from the view, you may need to delete it from your working folder.

If you add a new folder to a view, its working folder can be any of the following:

- ◆ Any folder on your workstation specified by you.
- ◆ A non-existing working folder specified by you and created by the application on your workstation. If the existing folder has child folders, one or more of them can also be added to the view.
- ◆ A child of the parent application folder's working folder. If you do not specify a working folder, the application appends the new folder's name to its parent's complete working folder path.

Note: If the parent folder's working folder path length exceeds the operating system's maximum working folder path length of 254 characters (including (\) backslashes), the application does not allow you to create the new working folder. Also, you cannot add a folder to a view if the parent folder is read-only.

The newly added folder assumes the parent folder's behavior, with a few exceptions. For example, the child folder might have the **Branch On Change** check box disabled because it makes no sense for this folder to branch.

New Folders

You can easily add folders to a project view. When a new folder is added:

- ◆ The working folder for the new application folder does not have to belong to the same hierarchy as the other application folders' working folders. However, if it uses the same drive letter as the root folder's working folder, its path is stored as a relative path based on the path to the working folder of its parent folder in the hierarchy.
- ◆ Its name can be different from the name of its working folder.
- ◆ If the new working folder has child folders, a folder can be created for each of the children. Essentially, the newly added folder becomes the root of a new branch of folders. The application folders created for the child folders take the names of their working folders—at least initially. The working folders retain their relationship to the working folder that is the root of their hierarchy (that is, the working folder for the newly added folder). If you change the path to the newly added folder's working folder, you also change the path to these working folders (unless you manually set an absolute path for these working folders).

StarTeam indicates folders on disk that do not map to a StarTeam folder with a **Not-In-View** icon . This indicates that you do not have the folder in the project view.

Existing Folders

You can add more folders to a view by moving them or sharing them from other views on the same server configuration. When a folder is moved or shared, it either keeps its absolute path or its relative path and is applied to its new parent folder. When a moved folder's path is relative, it usually ends up with a different working folder than it previously had. When a shared folder's path is relative, the shared folder has a different working folder in each location.

Note: The application does not allow you to create a working folder if a shared or moved folder's new working folder path exceeds the operating system's maximum working folder path length of 254 characters (including (\) backslashes).

Both the current view and the view from which the folders are moved or shared must use the same server configuration—and, therefore the same database and repository.

Related Concepts

[Overview of Folders and Paths](#)

Related Procedures

[Creating Projects](#)

[Working with Folders and Items](#)

Files

To place a file under version control, it must be added to a folder in a StarTeam project view, which stores a copy of the file in the StarTeam repository. After the file has been added to StarTeam, you and other members of your team can check it out, revise it, and check in new revisions, while StarTeam maintains information on all revisions of the file. Note that all check-ins in StarTeam are atomic.

When checking out a file revision, you should verify that you have the *tip* or latest version of the file. Doing this ensures that the file you see contains the latest changes. If you intend to modify the file, you should check it out with an exclusive lock, to indicate to others that you are working on it.

When you check a file in, StarTeam records the file changes as a new revision. As part of the check-in process, you can remove the lock, notifying others that the file is available, or maintain the lock, showing that you intend to continue working on the file. If two team members change the same text file simultaneously or if one member changes an outdated file, StarTeam contains a merge option that allows the file changes to be combined so that no work is lost. In such cases, StarTeam assigns a *Merge* status to the file.

Note: The SDK, StarTeam Server, and most clients support files larger than 4 GB. If you plan on taking advantage of large file support, you should upgrade all users to the current StarTeam client. Large file sizes are not compatible with older StarTeam applications.

Files Under Version Control

If a file resides in the working folder of an application folder, you can add that file to the application folder. This operation places that file under version control. A copy of the working file becomes the first revision of that file stored in the repository. If the working file is deleted later, the data is not lost because a copy exists in the repository. The application creates a new revision of this file in the repository every time you check the file in.

Every time you check a file revision out, its contents are copied to a working folder. Checking out a revision also ensures that you have the *tip* or a specific revision to work on. For example, you may need a team member's most recent changes to a file, or you may have deleted the working file from your hard drive and now need another copy.

The application enables you to label the *tip* revisions of every item within a view. For example, when the project reaches a particular milestone (such as beta), you might give the view's items a label, called a view label. Then you can configure the view to return to the way it was at the time the label was applied, check out revisions as a group using that label, create a new view based on the label, or assign the label to a promotion state.

The application also provides revision or version labels. You can label one or more revisions as you check them in or by applying the label to each of the revisions using the **Labels** command on the File menu. StarTeam makes it easy to check out those files as a group using the label. A file revision can have any number of labels. However, no two revisions of the same file in the same view can have the same label.

Recommendations for Working with Files Under Version Control

Here are some recommendations about using files under version control:

- ◆ To let other team members know that you intend to make changes to a file, change the lock status to *exclusive* as part of the check-out procedure.
- ◆ As part of the check-in process, you can notify others both that you are finished making your changes to the file and that it is available for them to check out by removing the lock status.
- ◆ If you intend to continue making changes to the file but still want to check it in for backup purposes, keep the file locked.
- ◆ If two team members change the same text file simultaneously or if one member changes an outdated file, you can use the merge option to combine the changes in these files so no work is lost. In such cases, the application gives the file a Merge status.

- ◆ To prevent yourself from changing a file that you have not locked, select the **Mark Unlocked Working Files Read-only** personal option. Then, if you check out a file that you have not locked, the working copy becomes read-only.

Related Concepts

[Overview of Folders and Paths](#)

[Atomic Check-ins](#)

Related Procedures

[Managing Files](#)

[Working with Folders and Items](#)

[Managing Labels and Promotion States](#)

[Comparing and Merging Files and Folders](#)

[Customizing Personal Options](#)

Change Requests

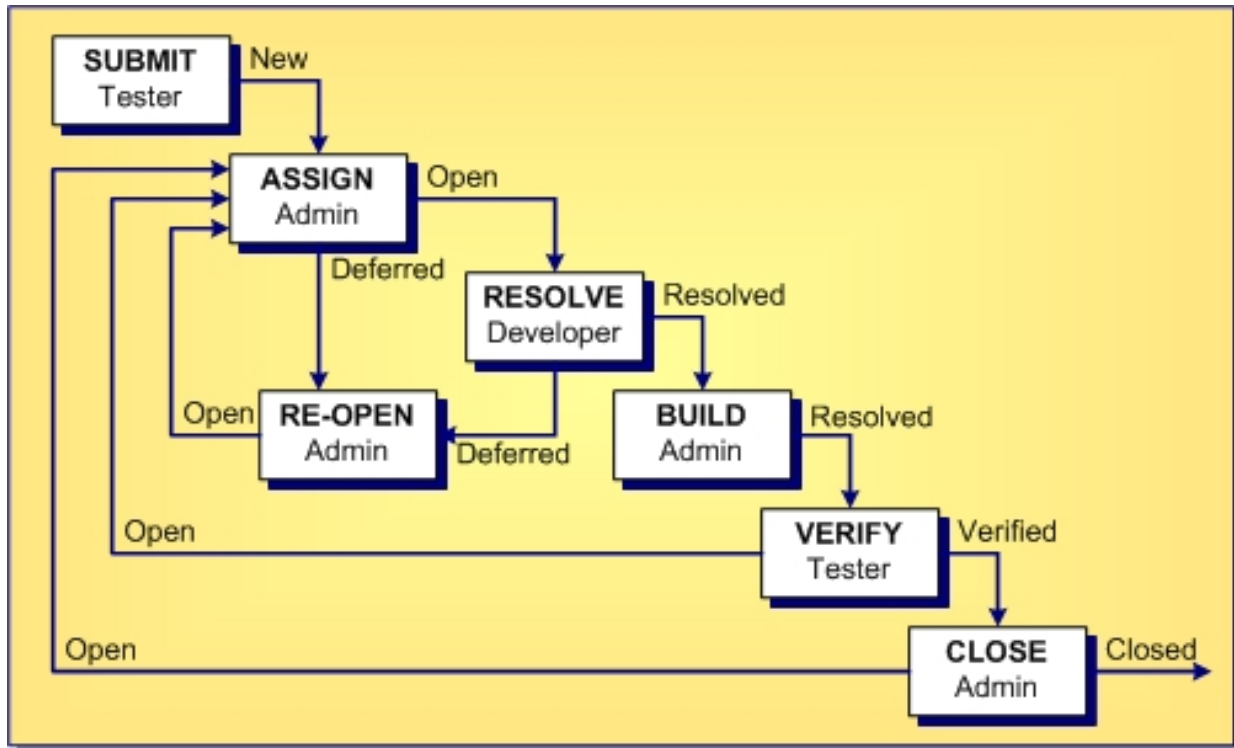
A change request is a request to change something within the scope of a project. For example, you might suggest a product enhancement or request a fix for an error or problem. To use the change request tracking system effectively, you need to understand the model on which it is based.

The change request component allows you to:

- ◆ Attach change requests to any folder. In the application, change requests can be attached to any project folder or shared among folders or other views in the same server configuration. You can also link a change request to any other item, such as a file. In many other defect tracking systems, a change request can be associated only with a project, even though it requires modification of a particular file.
- ◆ Save time when updating change requests. When you check in a file or group of files, you can indicate the change requests that are fixed by the files being checked in. This feature saves the time required to change the status of each change request separately.
- ◆ Make only appropriate status changes. When you create a change request, the status options are New, Open, Deferred or a resolution. The resolutions are Cannot Reproduce, As Designed, Fixed, Documented, and Is Duplicate. After resolution, a change request can only be verified or reopened. After verification, a change request can only be closed or reopened.
- ◆ Benefit from automatic changes based on the status of the change request. The application automatically changes the person responsible to coincide with the current status of the change request. When a change request is resolved, the responsibility for the change request automatically reverts to the person who entered the change request, who is usually the best person to verify its resolution. When a change request is reopened after being resolved, the responsibility is automatically set to the user who resolved it. If desired, you can override these automatic changes and make another person responsible.
- ◆ Base change requests on the build in which the change request is resolved. When a change request receives a **Fixed** or **Documented** status, the value of its **Addressed In Build** field becomes **Next Build**. When that build label is created, the application replaces **Next Build** with the name of the build label, letting testers know the build to use when verifying change requests.

Note: This help system explains how to use the standard property dialog to create and edit change requests. Depending on how your team has set up the application, you may see a different dialog called an alternate property editor (APE). Even if you use the standard property dialog for change requests, your company or team leader may implement change request guidelines that differ from those discussed in this help system.

Change Request Tracking System Model



The above diagram and steps show the change request tracking process. The boxes represent the steps taken from the time that the change request is submitted until the time it is closed. Each box indicates an action and the team member most likely to be responsible for performing this action. The arrows show the status of the change request at the time of each step.

The change request tracking system consists of the following steps:

Step 1: A team member creates a new change request that does either of the following:

- ◆ Summarizes a problem with the product and lists the steps taken to reproduce the problem.
- ◆ Suggests an enhancement to the product.

This change request has a status of *New*.

Step 2: Another person, such as an administrator or team leader, decides whether to fix the problem or add the suggested enhancement to the product. This person can:

- ◆ Set the status of the change request to *Open*, and assign a team member to resolve it.
- ◆ Set the status of the change request to *Deferred* because it is worthwhile but will not be done at this time.
- ◆ Set the status of the change request to *Is Duplicate* because this is not the first time it has been submitted. If desired, a link can be created between a change request and the original submission so that you can track the change request along with the original submission.
- ◆ Set the status of the change request to *As Designed* because the product is supposed to work this way, meaning there is no defect.

Change requests with an *Open* status go to step 3.

Step 3: The person assigned to resolving the change request changes the status of the change request to *In Progress*. Later on, after this person finishes examining the change request, he or she changes the status to one of the following:

- ◆ *Fixed*
- ◆ *Documented*
- ◆ *Cannot Reproduce*

Step 4: Next, a team member (usually a tester or quality assurance engineer) verifies the change request. For example, a test case may be developed to determine if the problem is really fixed, documented or not reproducible and changes the status to one of the following verified statuses:

- ◆ *Verified As Designed*
- ◆ *Verified Cannot Reproduce*
- ◆ *Verified Documented*
- ◆ *Verified Fixed*
- ◆ *Verified Is Duplicate*

Step 5: Finally, another team member changes the status to *Closed*. This person may then perform activities related to closing the change request, such as retesting the change request before closing it or adding it to a report to be included in the next release of the product.

In most of the above steps, the change request can be reopened and reprocessed.

Built-in Workflow for Change Requests

StarTeam has a built-in workflow for change requests that automatically sets many of the values associated with change requests. This built-in workflow determines these settings based on the setting of the *Status* field for the change request.

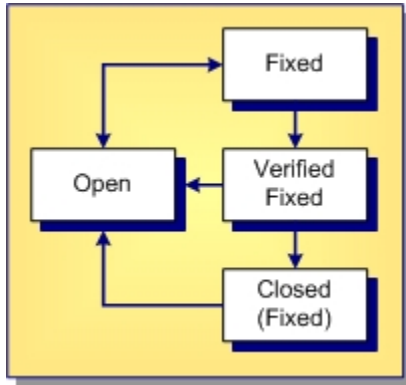
You cannot add additional settings to the *Status* field. However, you can rename them to better suit preferences set by your organization. For example, your organization may prefer to change the name of the Status *New* to *New Change Request*.

When you alter the status of a change request, the built-in workflow automatically selects the appropriate properties associated with the change in status.

After selecting *New*, *Open*, or *In Progress*, six new statuses display in the *Status* drop-down list box. These statuses, which are associated with the status you selected, are:

- ◆ *Deferred*
- ◆ *Cannot Reproduce*
- ◆ *As Designed*
- ◆ *Fixed*
- ◆ *Documented*
- ◆ *Is Duplicate*

Lifecycle for Change Requests



The above diagram shows the lifecycle for a change request with an initial status of *Open*. The status was then set to *Fixed*. After this setting, the built-in workflow added an additional status field of *Verified Fixed*. Finally, the change request was closed, meaning its status was set to *Closed (Fixed)*.

The diagram also shows that a change request can be reopened at any stage in its lifecycle because the arrows leading from each of the three fixed statuses can lead back to the *Open* status at any time.

Summary of the Change Request Automatic Workflow

The following table summarizes the steps used in processing change requests as explained in this topic. It includes the automatic workflow changes the application makes to change requests based on their statuses.

Change Request Process Management

Step	Description
Submit	<p>Anyone (usually a tester or quality assurance engineer) can submit a change request.</p> <p>Process: Select the Change Request tab. Then select New from the Change Request or context menu.</p> <p>A change request has the following default properties (which you can change if necessary).</p> <p>Status: New</p> <p>Severity: Low</p> <p>Priority: Not prioritized</p> <p>Type: Defect</p> <p>Platform: All</p> <p>Last Build Tested: Current build label</p> <p>Entered by: Person currently logged on to the application</p> <p>Many other fields are initially blank. Some team leaders prefer to have all change requests submitted at the root folder. They use drag-and-drop to move the change requests to the appropriate child folders.</p>
Assign*	<p>Process: The team leader finds all new change requests and does one of the following:</p> <ul style="list-style-type: none">■ Opens the change request and assigns it to a developer, help writer, or other appropriate team member.■ Defers the change request until a later date, perhaps the next release of the product.■ Specifies that the change request is <i>As Designed</i> and not to be fixed. If the change request status is <i>Open</i>, no automatic changes occur. If the change request status is <i>Deferred</i> or <i>As Designed</i>, then <i>Addressed in Build</i> is disabled and the responsibility is assigned to the user who created the change request.
Resolve	<p>Process: Users find the <i>Open</i> or <i>In Progress</i> change requests assigned to them, and do one of the following for each request:</p>

- Resolve the problem in the system and update the properties of the change request. (The statuses that indicate that a change request has been resolved are *Cannot Reproduce*, *As Designed*, *Fixed*, *Documented*, or *Is Duplicate*.)

- Defer the change request until a later date, perhaps the next release of the product. Your team leader may prefer that you do not defer change requests.

If the change request status is one of the possible resolution statuses, then *Addressed in Build* becomes *Next Build* for *Fixed* and *Documented* statuses. It becomes disabled for other statuses. By default, the responsibility is assigned to the person who submitted the change request, who is expected to verify the resolution.

If the change request status is *Deferred*, then *Addressed in Build* is disabled and the responsibility is assigned, by default, to the user who created the change request.

Build Who builds the project? The project view may have a formal or informal build process. However, at some point, all the files, and so on currently in the view receive that build label. It is usually applied to the source code files, and so on that were compiled (and may need to be changed) rather than to the executable files that result from the build.

Effect on change requests: For any resolved change request that has *Next Build* as the setting for its *Addressed In Build* property, *Next Build* is replaced with the next build label that is created.

Note: If a new build label is based on a past configuration (rather than the current configuration), it has no effect on the *Addressed In Build* property.

If a change request has not branched in its current location, *Next Build* may be replaced with a build label from another view. For example, suppose you create a branching child view or share a folder from one view to another. Suppose that *Next Build* is the value of some change request's *Addressed In Build* property and that change request has not branched. When a build label is created in the source view, *Next Build* is replaced with the name of that build label, regardless of the location.

Verify* The person who submitted the change request (usually a tester or quality assurance engineer) verifies a resolution.

Process: Install the build in which the resolution is to be verified and determine whether the change request has been resolved correctly. Do one of the following:

- Verify the change request, marking it as *Verified Cannot Reproduce*, *Verified As Designed*, *Verified Fixed*, *Verified Documented*, or *Verified Is Duplicate*.

- Reopen the change request and update the setting for *Last Build Tested*.

If the change request status is *Verified*, no automatic changes occur.

If the change request status is *Open*, *Addressed in Build* is blank. If the change request has changed from resolved to *Open*, the user who changed the status to *Fixed* or *Documented* becomes responsible.

Close* Usually the team leader closes the change request.

Process: The team leader does one of the following:

Reviews and closes the verified change request.

Reopens the change request.

If the change request status is *Closed*, then no automatic changes occur.

If the change request status is *Open*, then *Addressed in Build* is blank. If the change request has changed from resolved to *Open*, the user who changed the status to *Fixed* or *Documented* becomes responsible.

If the status of a change request changes from *Verified* to *Open*, the user who changed the status to *Fixed* or *Documented* becomes responsible and *Addressed in Build* is blank.

*Changes in status can result in automatic changes to other properties.

Related Procedures

[Working with Change Requests](#)

Requirements

Requirements are supported for the Enterprise Advantage license and display in the Requirement tab of the upper pane in the clients. With the Requirement component, you can create requirements within the application and show the dependencies among them. For example, if one requirement must be fulfilled before a second requirement can be fulfilled, the first can be made a child of the second. If your company enforces process rules, the requirements you establish can also be used to drive the development process. Administrators and other authorized users can publish requirements from CaliberRM to StarTeam using Publisher to StarTeam, which is delivered with CaliberRM.

Requirement Characteristics

The requirements in the upper pane have the following characteristics:

- ◆ They are attached to the folder selected from the folder hierarchy.
- ◆ They match the filter selected from the Filter drop-down list box.
- ◆ They match the depth specified by All Descendants. (You can click the button on the toolbar or select All Descendants from the Requirement menu.)

Note: Icons display to the left of a requirement in the upper pane to indicate its status and whether you have read the latest revision.

How Requirements Can Help

By using a requirements-driven development processes, companies can prevent consuming, costly misunderstandings and shorten time to market. To accomplish this, you can use the StartTeam built-in Requirement component as your basic tool, or publish complex requirements to StarTeam from Borland CaliberRM. Using requirements enables business analysts, managers, developers, QA staff, and others to:

- ◆ Organize business, user, and functional requirements in a hierarchical format.
- ◆ Indicate the dependencies among requirements.
- ◆ See all layers of requirements at all times.
- ◆ Prioritize requirements by importance.
- ◆ Identify the impact of changes to requirements.
- ◆ Use requirements to estimate work.
- ◆ Identify the person creating the requirement.
- ◆ Notify those who will be responsible for fulfilling the requirements.
- ◆ Track the requirement lifecycle from submitted to completed or rejected.
- ◆ Provide requirements with a context by linking them to files, change requests, and topics.

Related Procedures

[Using Requirements](#)

Tasks

The Task component allows the creation of task lists and work assignments. As a standalone, the Task component is very useful for managing a project. It allows team members to indicate who should do what and when, see current task status, estimate hours required to complete a task, record hours spent completing the task, and compare estimated to actual times. Because the application contains both a version control system and a change request system, it also allows tasks to be linked to the files and product defects or suggestions with which they are associated.

The Task component can be used independently or interoperate with data from Microsoft Project. It can display tasks in a tree format, which clearly shows the relationship between tasks and subtasks, or in a list format, which allows tasks to be sorted, grouped, or queried, or specific fields to be selected for display. To improve efficiency, each task displays icons that identify its status, priority, milestone, and need for attention. For information about interoperating with Microsoft Project, see the StarTeam Microsoft Project Integration User's Guide.

With the StarTeam Task component, you can create an individual task or a summary task that has a set of subtasks. It is recommended that you plan tasks before entering them because:

- ◆ A task that has even one subtask cannot have work records added to it, although work records can be added to subtasks. The application assumes that the name of the task indicates a goal, perhaps a milestone, that will be reached when the subtasks are completed.
- ◆ After a work record has been added to a task, you cannot create subtasks for it.

Note: Regardless of whether work records can be added to a task, you should assign the responsibility for its completion to a specific team member. If work records can be added to a task, you should also estimate how long the task should take.

Related Procedures

[Using Tasks](#)

Topics

Topics are threaded conversations — that is, a series of messages that indicate how the messages are related. Each series of messages forms a tree with the initial message at its root. The Topic component provides threaded conversations that you can place in specific project folders and link to specific project items. For example, you can link a topic to the change requests and file revisions that result from the topic discussion.

The upper pane of the client consists of topics and a series of responses to each topic. A series of topic trees are eventually formed, each of which consists of a root topic and its responses. The topic tree resembles a conversation that may go on among several people. In the client, this is called a threaded conversation because a topic and its responses are threaded together, starting with the root topic. By reading each response in a thread, one after the other, and the responses to those responses, you can see how the discussion has evolved. A number of other operations can be performed on topics or responses such as moving or sharing them.

Historical Value of Topics

Topics can raise general questions about the project or start very specific discussions about issues, such as feature implementation. While the responses can lead to resolution of these issues, the historical value of these conversations to the project can be even more significant. Future team members can:

- ◆ Reassess decisions more capably
- ◆ Avoid retrying solutions that were previously found faulty
- ◆ Understand why a particular solution to a problem became necessary and, therefore, not replace that solution with one that does not meet all the necessary criteria

How Topics Can Help

Any type of threaded messaging improves teamwork on product development. However, because StarTeam has tightly integrated components, it enables team members to:

- ◆ Search topics and responses for specific words or phrases
- ◆ Sort topics and responses
- ◆ Filter topics and responses
- ◆ View relationships between topics and their responses
- ◆ Move and share topics (from the tree format)
- ◆ Link topics directly to folders or other items, such as change requests.
- ◆ Ask questions and quickly receive input while working on a file.
- ◆ Attach notes to a topic explaining why a particular method was used.
- ◆ Point out aspects of the project that may need to change in a later release.

Related Procedures

[Using Topics](#)

[Working with Folders and Items](#)

Links: Internal and External

A link is a connection between two folders, two items, or a folder and an item on the same server, or on two different servers (called *External Links*). You can view links by using the Link tab in the lower pane of the client. Creating links can have a number of benefits. For example, when you are reviewing a file and the change requests that affect that file have been linked to it, you can review the change requests without selecting the Change Request tab.

In addition, linking files to change requests enables you to mark the change requests as fixed when you check in the corresponding files. In turn, if you link each set of files to the requirements document that the files fulfill, you can easily refer to or update the document. For more information on linking and unlinking items, refer to the links at the end of this topic.

A link does not provide a connection to a single share (or reference), but to all related shares and branches of an item. Links are not affected by any item operations, such as branching, moving, sharing, and so on. By default, a link connects the tip revisions of the linked pair.

Links can either be pinned or floating:

- ◆ Pinning a link means that you can lock the link to the tip revision. The context menu in the link tab enables you to pin links to the source or target items or both.
- ◆ Floating a link means that you allow the link source or target to change from tip revision to tip revision as new revisions are created. The context menu in the link tab enables you to float links to the source or target items or both.

Links, as with all other items, have context menus in their tabbed panes which allow you access to more information about the item.

Related Concepts

[Check-in and Check-out Operations](#)
[Process Items and Process Links](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

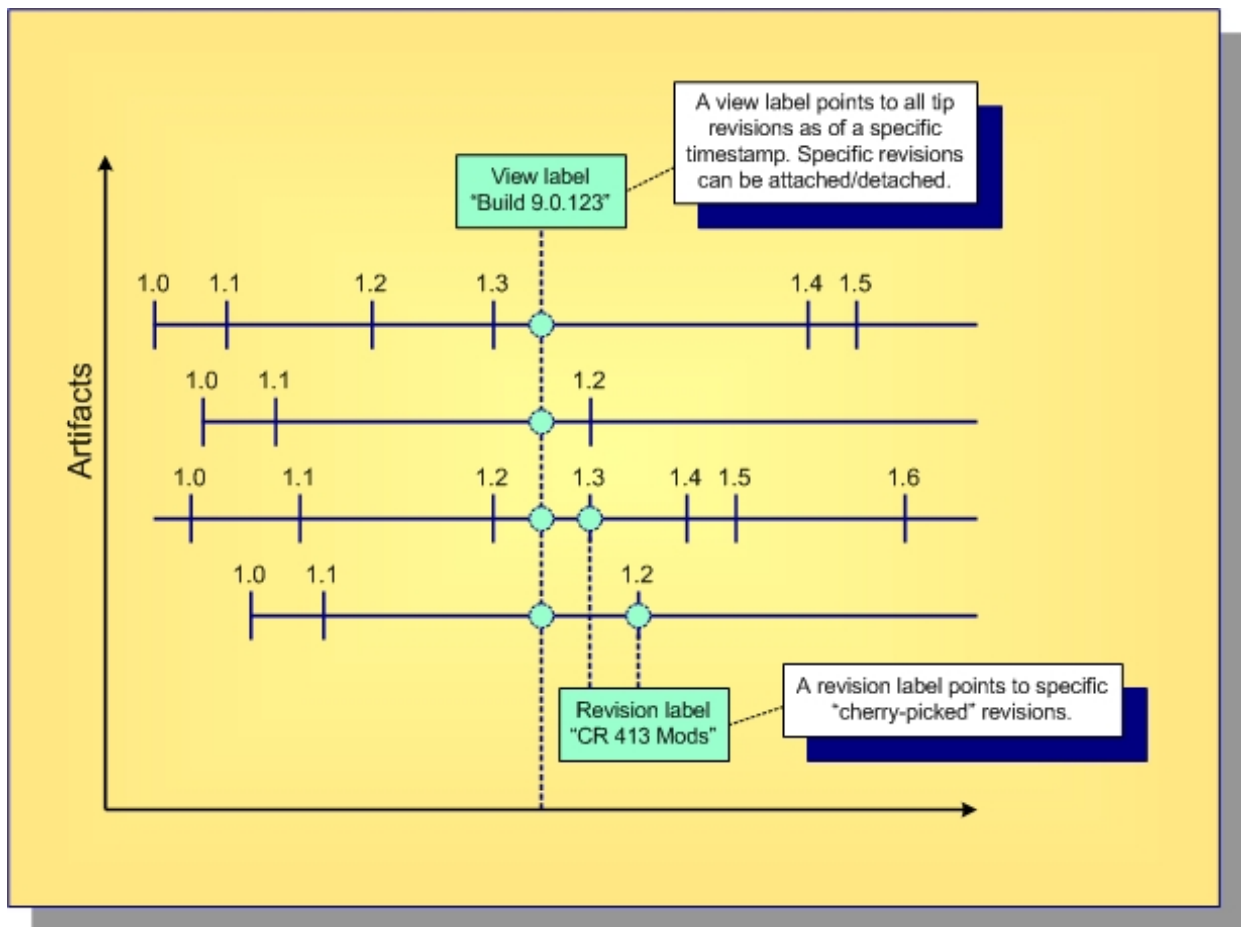
Labels

Labels provide a way to identify a specific set of artifact revisions for any purpose. Unlike containers such as views or folders, we say that specific artifact revisions are attached to labels instead of being contained by them. Furthermore, a label is usually fairly static—the revisions attached to it are not expected to change much, if at all. A label provides a lightweight snapshot.

Here is what is important to know about labels:

- ◆ **View scope:** Labels “belong” to a view. That is, each view has its own set of labels. This also means that every label has a *name* that must be unique from other labels belonging to the same view. Each label also has a *description* that helps users understand the purpose of the label.
- ◆ **Frozen labels:** A label can be frozen, which means no new artifacts can be attached to it, and attached artifacts cannot be detached nor reattached at a different revision. Conversely, non-frozen labels can have all of these modifications. Since many organizations depend on the immutability of frozen labels, a specific security permission is required to freeze or unfreeze a label.
- ◆ **Labels attach to items:** StarTeam actually attaches specific *item revisions* to a label, each of which infers a specific *artifact revision*. Since each item specifies a parent folder, the artifact is attached in a specific folder. This means that if you move an item, you need to reattach it to any label for which you want to reflect the artifact in its new folder. If you don’t reattach a moved item to an existing label, it will continue to be attached to the label in its old folder (which may be what you want).
- ◆ **View labels:** StarTeam supports two kinds of labels. A *view label* is inclusive of the entire view as of a specific timestamp. That is, when a view label is first created, it is attached to every artifact revision that was tip as of that timestamp. (A snapshot of “now” can be used as the timestamp.) Don’t worry; this is not as expensive as it sounds: StarTeam view labels are cheap. After a view label is created, new items can be attached to it, attached items can be adjusted to be attached at a different artifact revision, and items can be completely detached from it. A view label can be marked as a *build label*. New build labels are automatically assigned to all CRs whose **Addressed In** field has the value **Next Build**.
- ◆ **Revision labels:** The other kind of label StarTeam supports is a *revision label*. Revision labels are exclusive in that, when you first create them, nothing is attached to them. After you create a revision label, you attach specific items, building it up to reflect a specific set that is typically a small subset of the view. StarTeam can automatically attach new file revisions to a revision label at check-in time if you like.
- ◆ **Label cloning:** Even if frozen, both view and revision labels can be cloned. That is, you can create a new label starting out identical to an existing label and then adjust the revisions attached to it. A common practice is to clone a previous label, attach only new file revisions that were created to fix a bug, and use the new label to identify the file revisions for a new build candidate, or release candidate, or whatever.

Example view and revision labels are shown below.



Note: Not all items in a view have to be attached to a label. Conversely, an item can be attached to any number of view and revision labels as you like, but only one revision of an item can be attached to any specific label.

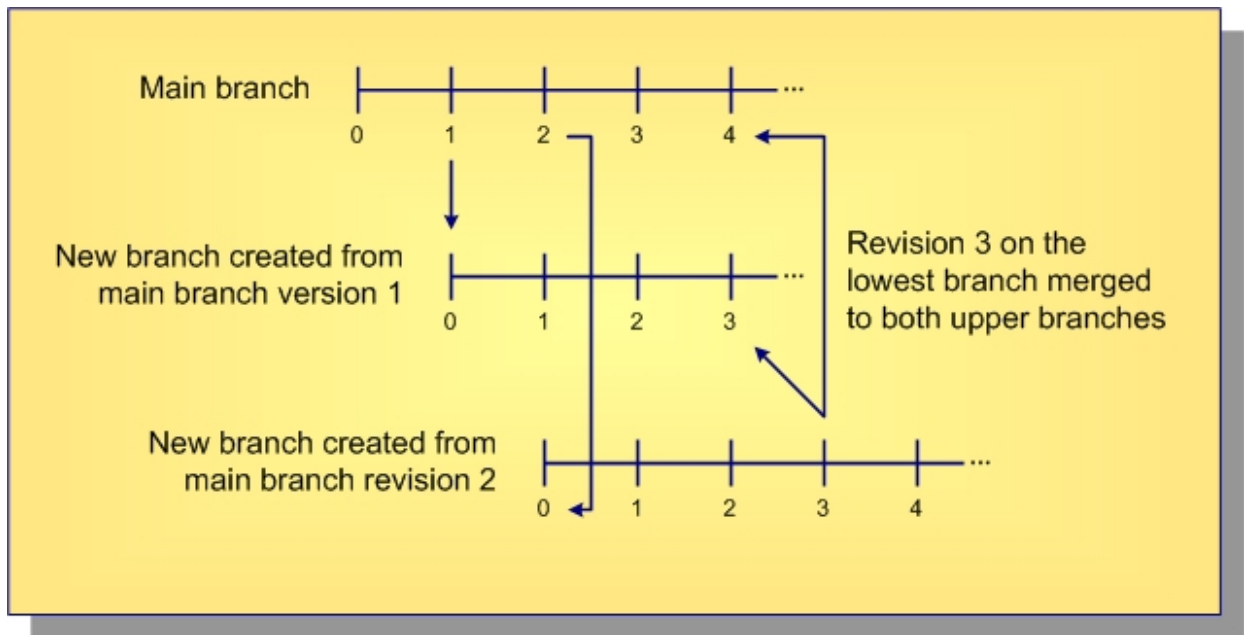
Branching, Merging and Dot Notation

This topic explores the concepts of branching, merging and dot notation as they relate to StarTeam.

- ◆ **Branching:** Allows an artifact's version stream to be forked. Each fork can be independently modified, receiving its own versions.
- ◆ **Merging:** Propagating a change on one branch to another branch.
- ◆ **Dot Notation:** A dotted decimal notation assigned to artifact revisions to indicate both the branch on which the revision resides and the relative version number of the revision within the branch (for example: 1.4 or 1.2.1.5).

Branching

There's a big difference between *copying* an artifact and *branching* an artifact. Although copying allows each copy to be modified independently, StarTeam does not know that copied artifacts are related to each other in the repository. In contrast, with branching, StarTeam retains special knowledge about an artifact's branches. This information supports things such as intelligent revision comparison and *three-way merging*, which is discussed below.



When a new artifact is added to the repository, a *main branch* is started and new revisions are added to it. At any time, a parallel "child" branch can be started. In the example above, one child branch is created from main branch version 1, and another child branch is created from main branch version 2. Within a branch, the version number starts over (at zero in this example). New revisions are applied to a specific branch, incrementing the version number on that branch but not affecting other branches. Branching is needed to support parallel development on files. However, there are advantages to allowing non-file artifacts to branch as well. For example, if a defect artifact can be branched, the two branches can be used to track fixes to the same defect that exist in different releases.

Merging

Inevitably, a change on one branch will need to be propagated to another branch. In the diagram above, version 3 of the artifact on the lowest branch is applied to both of the parent branches. However, you can't just copy a revision

from one branch to another branch—this could wipe out changes that are specific to the target branch. Instead, what you want to do is *merge* the changes from the source to target branch.

Note: *Overwriting* the target artifact with the source revision instead of merging it is sometimes the desired effect, for example with binary files.

More specifically, StarTeam stores synchronization information that allows *three-way merging*. The three parts of a merge operation are (1) the *source revision* containing changes to be propagated, (2) the *target revision* that will be modified, and (3) the last source revision common to the source and target branches, known as the *common ancestor*.

For files, merging is done by passing these three file revisions to the File Compare Merge tool. The tool compares both the source revision and target revision to the common ancestor revision and determines two important things: (1) what changes appear in the source revision only that should be propagated to the target revision, and (2) what changes appear in the target that may conflict with changes in the source revision.

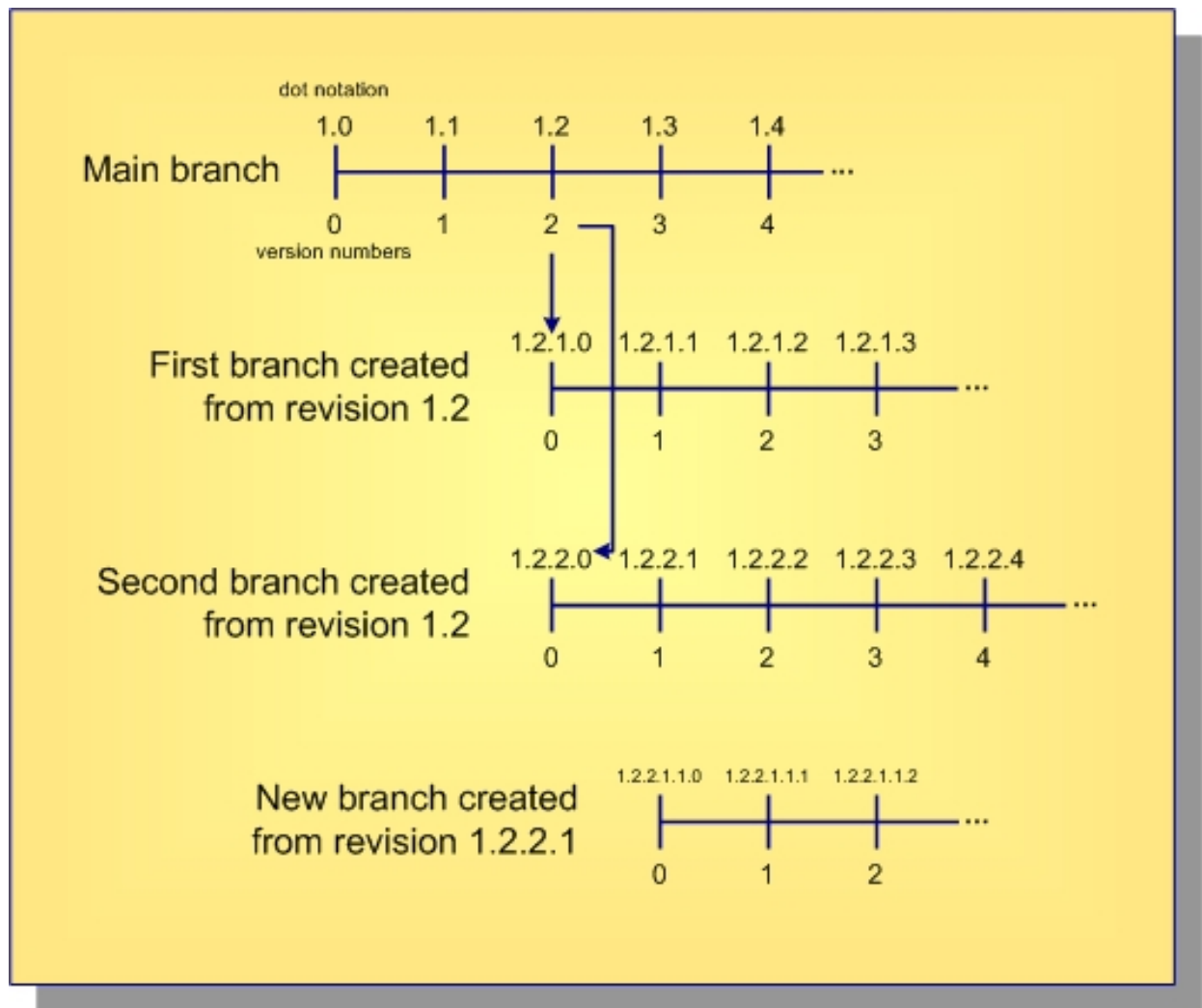
In many cases, merging detects no conflicts, so the File Compare Merge tool automatically propagates the source changes to the target revision. When conflicts are detected (and sometimes even when none are), the File Compare Merge tool displays the differences to a user who can review the differences, resolve conflicts, and approve the final result. The File Compare Merge tool then creates a result file reflecting the target file updated with changes. StarTeam adds the result file to the target revision's branch, creating a new revision.

In the diagram above, revision 3 on the lower child branch was merged to both of the upper branches. When it was merged to the main branch revision 3, it created main branch revision 4, which contains the merge results. (The arrow points to the revision that was created as a result of the merge.) For this merge, the common ancestor between the two branches is main branch revision 2: it is the most recent revision that both branches had in common. When the lower child revision 3 is merged with upper child branch revision 2, upper child branch revision 3 was created. For this merge, the main branch revision 1 is the common ancestor.

For files, there is more to merge than *contents*: files have other properties such as *name* and *description*. In order to propagate a name change, for example, merging a file requires merging these properties as well. Non-file artifacts that branch also require merging in order to propagate changes.

Dot Notation

In addition to a version number, StarTeam assigns each revision a dotted-decimal value called a *dot notation*. Whereas the version number is unique within a *revision branch*, the dot notation value is unique within the entire *revision tree*. An example is shown below.



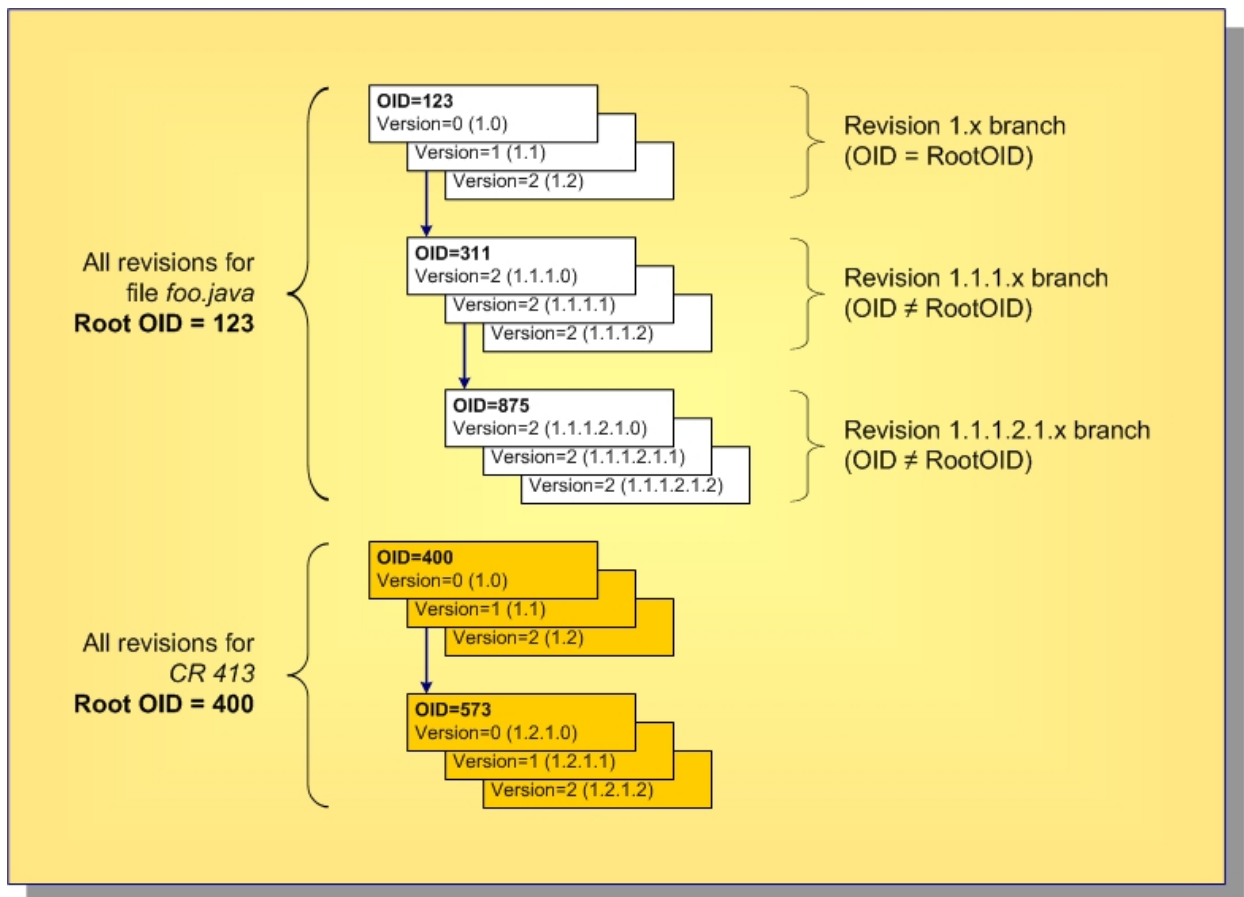
As shown, revisions on the artifact's *main branch* use the single dot notation pair $1.n$, where 1 indicates that it is the initial (first) branch and n is the same as the version number. When the artifact is branched from the main branch, revisions on the child branch use the dot notation $1.m.1.n$, where m is the main branch version number from which the branch was created and n is the version number on the new branch.

Note that an artifact can branch more than once from same point: in the example above, branches $1.2.1.n$ and $1.2.2.n$ were both created from main branch revision 1.2. The second 2 in $1.2.2.n$ tells you that this is the second branch from revision 1.2. Branch $1.2.2.1.1.n$ has three pairs of numbers, telling you that it is a third-level branch, created from parent revision 1.2.2.1.

Artifacts that can't branch (tasks, topics, and requirements) are always on the main branch, so their dot notation is always $1.n$.

Object IDs and Root Object IDs

All revisions in the same revision tree have the same *root object ID* (root OID). All revisions that belong to the same branch have the same *object ID* (OID). Furthermore, for all revisions on the main branch, the object ID and root object ID are the same. This is illustrated below.



In this example, the file `foo.java` started with OID and root OID 123, and the corresponding 1.n branch has revisions up to 1.2. At revision 1.1, it branched to form the 1.1.1.n branch, which uses the new OID 311. Revision 1.1.1.2 was branched to form the 1.1.1.2.1.n branch with OID 875. But all revisions in the entire branch tree have the original root OID 123. Each revision holds the properties specific to it: name, description, contents, etc.

Also shown is a change request (CR 413) that began with OID and root OID equal to 400. At revision 1.2, it branched to form branch 1.2.1.n with OID 573.

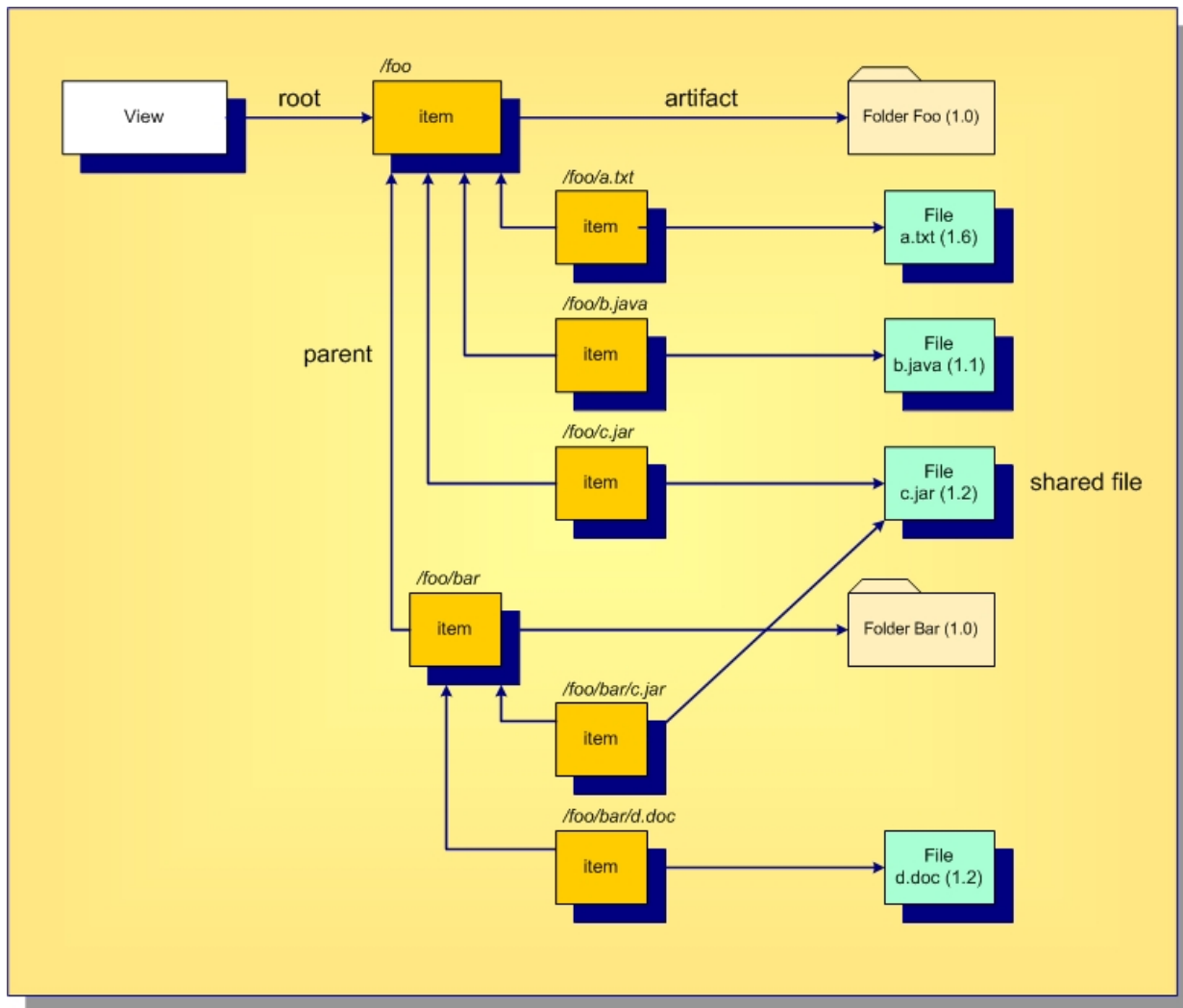
Now, consider a *folder* artifact. Each revision holds properties such as *name*, *description*, and *exclude spec* (file patterns to ignore within working folders). What's really different about StarTeam folder artifacts is that they do not have a property that represents their contents.

Sharing and “Cheap Copies”

Over time, you will have a lot of artifacts—especially files—and some files will have a lot of branches. Consider the effect of “containers”: if you have a lot of teams, software components, and releases, you will need a lot of independent projects, subprojects, or other containers to support parallel development and separate maintenance. Often the same files will be needed in each of these containers. How do you get the files you need to each of these containers? Forcing every file to branch in order to get a unique branch in every possible container could be a lot of branching, which is expensive.

StarTeam systems addresses this problem with a technique known as *cheap copies*. This involves creating references to files in a new container. Similar to UNIX links, this happens without actually copying the files themselves (that is, their content or their history). Unlike UNIX links, however, the first time a file is modified via a new reference, it is branched. For this reason, cheap copies are also referred to as “copy on write” sharing. Cheap copies support efficient branching with large projects.

In StarTeam, the folder hierarchy and the contents of each folder are specific to each view. Artifacts can belong to (or more properly be exposed through) any number of views and projects. Items are objects that select specific artifacts, connect them to a specific view, and organize them into a hierarchy. The diagram below shows how this works.



Every view has a *root item*, which always points to a folder artifact. In this example, the root folder name is *foo*. We can make any artifact in the repository belong to this folder by creating an item that points to the artifact we want and the root item as the *parent*. In this example, the files *a.txt*, *b.java*, and *c.jar* and the folder *bar* are all child elements

of the root folder *foo*. As you can see, the concept of *path name* is formed by concatenating the names referenced by the item structure. In this view, there is a file whose path name is */foo/bar/d.doc* because we can get to this artifact via the item path: folder *foo* to folder *bar* to file *d.doc*. If we want to change the folder in which *d.doc* appears, we change the *parent* of its associated item—the artifact itself is not modified. Notice that two items reference the file *c.jar*. This means that this file is contained in two different folders. We say that the file is *shared* in two places. This is analogous to UNIX links that reference the same file, causing it to appear in multiple directories. Sharing allows **any** artifact to be shared in multiple places. Since artifacts are “heavy” (they contain all the properties) and items are “light”, this is how “cheap copies” are made: we just create items pointing to existing artifacts.

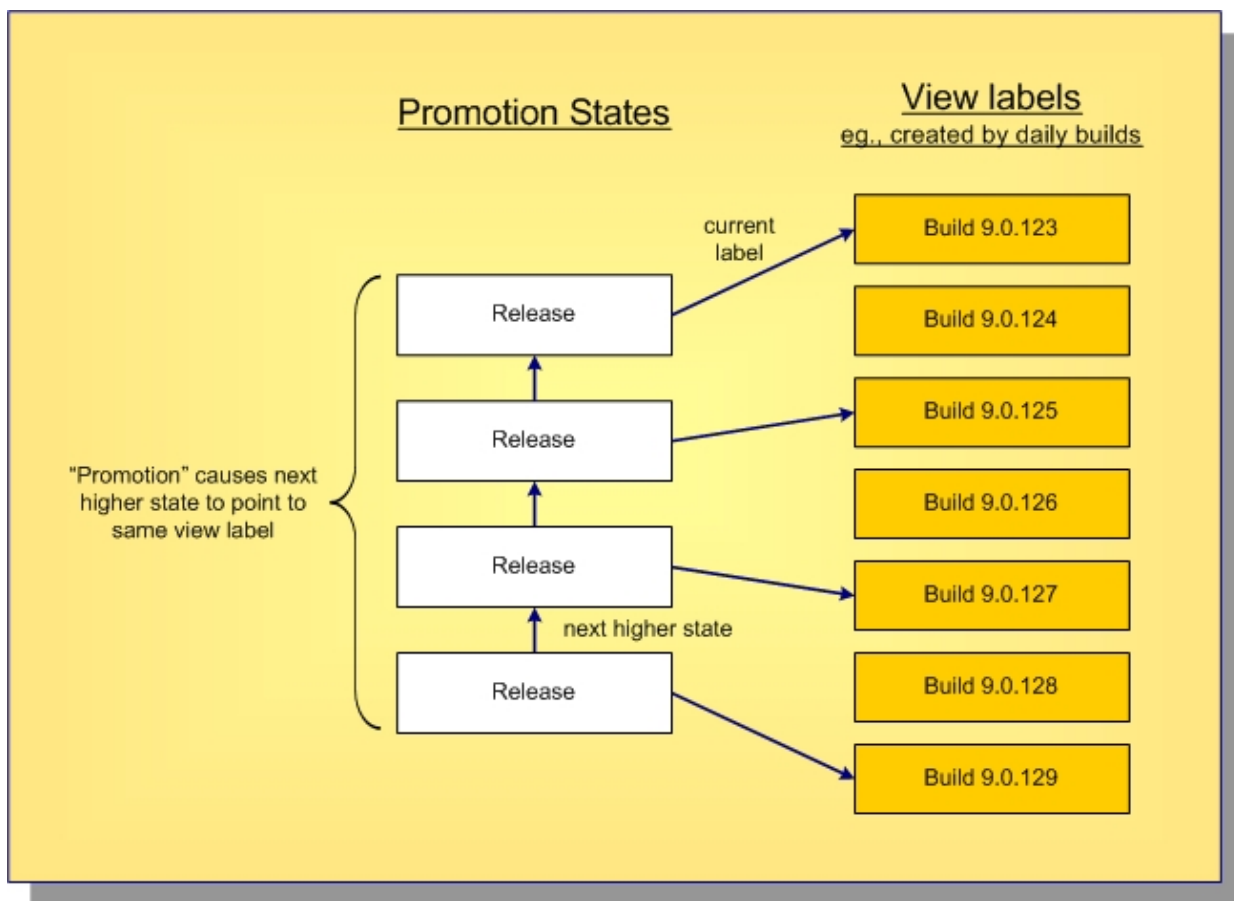
Promotion States

Promotion states provide an “intra view” change management facility. Promotion states are built on top of view labels, providing an ordered set of states through which items can be promoted within a view. Promotion states are generally used to move the entire view (or most items within it) through a series of steps based on passage of specific verification tests.

After defining a set of promotion states and what view label each state is initially mapped to, you then periodically create new view labels to represent specific view states (such as daily build candidates). Depending on your process, you then typically map the lowest-level state to the new view label and launch the first verification test. After the tests for that state complete, you “promote” it, causing the next higher state to be mapped to that view label. (Multiple states often point to the same view label.) When the tests for the final or “top most” promotion state passes, the view is ready for release, deployment, or whatever your process calls for.

Promotion states allow you to create build scripts, unit test scripts, deployment scripts, and so forth that operate on a specific promotion state without having to be modified to know about new view label names.

An example set of promotion states is shown below.



Testing and Reporting

The topics in this section provide getting started information about the StarTeam features and concepts used by a quality assurance engineer.

In This Section

[Testing and Reporting Overview](#)

Provides an overview of getting started concepts used in testing and reporting.

[Change Requests](#)

Describes the change request component and its built-in workflow.

[Queries](#)

Describes the purpose for using a query.

[Reports](#)

This topic provides scenarios for generating reports and information about customizing report templates.

[Charts](#)

Describes the charting features available in StarTeam.

[Filters](#)

Describes the purpose of a filter.

[Fields](#)

Describes fields used in the StarTeam clients.

[Data Export with Datamart](#)

Describes the chapter content in a sentence; used for part descriptions only.

[Cross-Server Configuration/Project Searches with Borland Search](#)

Describes the Borland Search product and where to find its associated documentation.

Testing and Reporting Overview

As a quality assurance engineer, you may perform some or all of the following actions:

- ◆ Enter and resolve change requests
- ◆ Update the status of change requests
- ◆ Generate charts and reports based on change request data
- ◆ Extract server data using StarTeam Datamart
- ◆ Create custom queries and filters for displaying change request data
- ◆ Version test plans
- ◆ Delete change requests

Change Requests

The Change Request component provides a defect tracking system that allows you to record defects in products, projects, or services and suggest possible enhancements.

Charts and Reports

StarTeam offers a wide variety of charts. You can filter out data in the upper pane of the client to display only the data that you want to include in your chart. In addition, you can select specific items from the filtered data to include in your chart. Charts are created from the data displayed (maximum of 60 fields) in the upper pane. You can use charts in a number of different ways. For example, you can use charts to track the number of closed and newly-opened change requests during a time period of a product development cycle.

StarTeam also offers a wide variety of pre-formatted, modifiable HTML report templates. Report generation is affected by sorting, grouping, and selecting items in the file, change request, topic, or task list of the upper pane.

StarTeam Datamart

StarTeam Datamart is a complementary product to the StarTeam Server. StarTeam Datamart uses the StarTeam SDK to communicate with the StarTeam Server to create a reporting database that you can use with popular third party reporting applications such as Crystal Reports and Business Objects (reporting applications are not included with StarTeam Datamart).

Related Concepts

[Change Requests](#)

[Queries](#)

[Reports](#)

[Charts](#)

[Filters](#)

[Fields](#)

[Data Export with Datamart](#)

[Cross-Server Configuration/Project Searches with Borland Search](#)

Related Procedures

[Working with Change Requests](#)

[Querying Data](#)

[Filtering Data](#)

[Creating Reports and Exporting Data](#)

[Creating Charts](#)

Change Requests

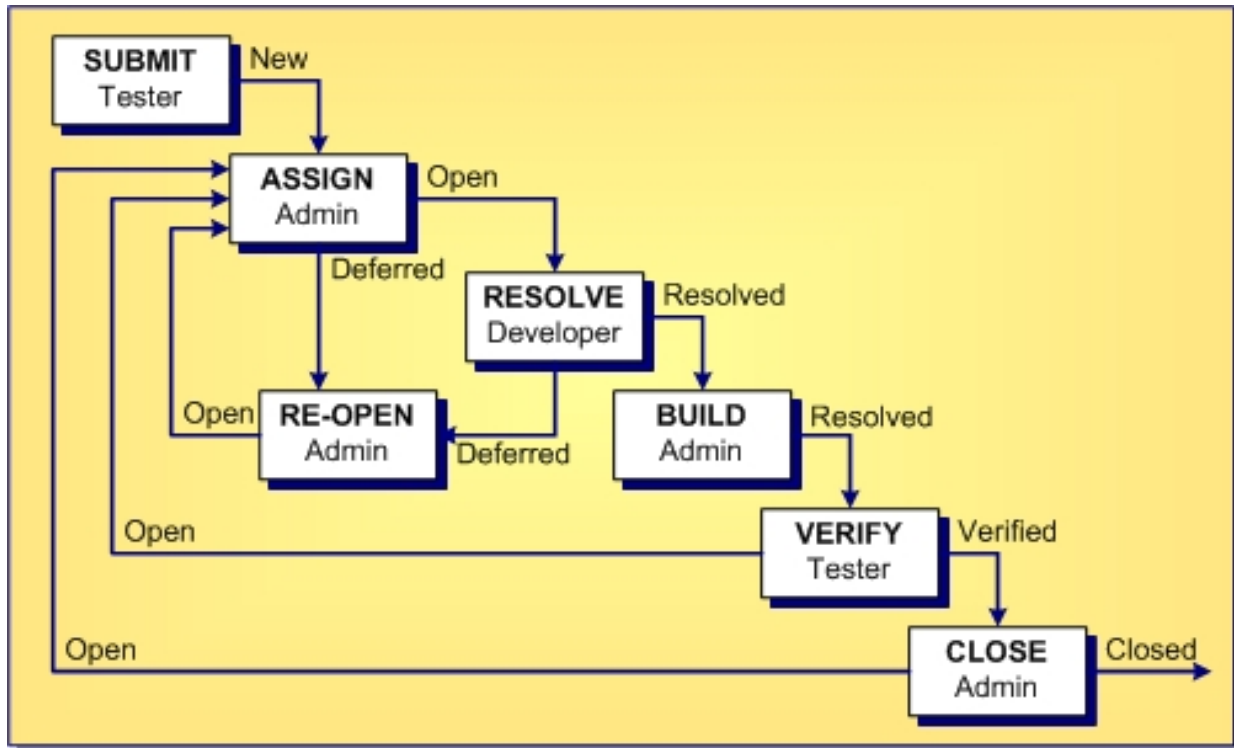
A change request is a request to change something within the scope of a project. For example, you might suggest a product enhancement or request a fix for an error or problem. To use the change request tracking system effectively, you need to understand the model on which it is based.

The change request component allows you to:

- ◆ Attach change requests to any folder. In the application, change requests can be attached to any project folder or shared among folders or other views in the same server configuration. You can also link a change request to any other item, such as a file. In many other defect tracking systems, a change request can be associated only with a project, even though it requires modification of a particular file.
- ◆ Save time when updating change requests. When you check in a file or group of files, you can indicate the change requests that are fixed by the files being checked in. This feature saves the time required to change the status of each change request separately.
- ◆ Make only appropriate status changes. When you create a change request, the status options are New, Open, Deferred or a resolution. The resolutions are Cannot Reproduce, As Designed, Fixed, Documented, and Is Duplicate. After resolution, a change request can only be verified or reopened. After verification, a change request can only be closed or reopened.
- ◆ Benefit from automatic changes based on the status of the change request. The application automatically changes the person responsible to coincide with the current status of the change request. When a change request is resolved, the responsibility for the change request automatically reverts to the person who entered the change request, who is usually the best person to verify its resolution. When a change request is reopened after being resolved, the responsibility is automatically set to the user who resolved it. If desired, you can override these automatic changes and make another person responsible.
- ◆ Base change requests on the build in which the change request is resolved. When a change request receives a **Fixed** or **Documented** status, the value of its **Addressed In Build** field becomes **Next Build**. When that build label is created, the application replaces **Next Build** with the name of the build label, letting testers know the build to use when verifying change requests.

Note: This help system explains how to use the standard property dialog to create and edit change requests. Depending on how your team has set up the application, you may see a different dialog called an alternate property editor (APE). Even if you use the standard property dialog for change requests, your company or team leader may implement change request guidelines that differ from those discussed in this help system.

Change Request Tracking System Model



The above diagram and steps show the change request tracking process. The boxes represent the steps taken from the time that the change request is submitted until the time it is closed. Each box indicates an action and the team member most likely to be responsible for performing this action. The arrows show the status of the change request at the time of each step.

The change request tracking system consists of the following steps:

Step 1: A team member creates a new change request that does either of the following:

- ◆ Summarizes a problem with the product and lists the steps taken to reproduce the problem.
- ◆ Suggests an enhancement to the product.

This change request has a status of *New*.

Step 2: Another person, such as an administrator or team leader, decides whether to fix the problem or add the suggested enhancement to the product. This person can:

- ◆ Set the status of the change request to *Open*, and assign a team member to resolve it.
- ◆ Set the status of the change request to *Deferred* because it is worthwhile but will not be done at this time.
- ◆ Set the status of the change request to *Is Duplicate* because this is not the first time it has been submitted. If desired, a link can be created between a change request and the original submission so that you can track the change request along with the original submission.
- ◆ Set the status of the change request to *As Designed* because the product is supposed to work this way, meaning there is no defect.

Change requests with an *Open* status go to step 3.

Step 3: The person assigned to resolving the change request changes the status of the change request to *In Progress*. Later on, after this person finishes examining the change request, he or she changes the status to one of the following:

- ◆ *Fixed*
- ◆ *Documented*
- ◆ *Cannot Reproduce*

Step 4: Next, a team member (usually a tester or quality assurance engineer) verifies the change request. For example, a test case may be developed to determine if the problem is really fixed, documented or not reproducible and changes the status to one of the following verified statuses:

- ◆ *Verified As Designed*
- ◆ *Verified Cannot Reproduce*
- ◆ *Verified Documented*
- ◆ *Verified Fixed*
- ◆ *Verified Is Duplicate*

Step 5: Finally, another team member changes the status to *Closed*. This person may then perform activities related to closing the change request, such as retesting the change request before closing it or adding it to a report to be included in the next release of the product.

In most of the above steps, the change request can be reopened and reprocessed.

Built-in Workflow for Change Requests

StarTeam has a built-in workflow for change requests that automatically sets many of the values associated with change requests. This built-in workflow determines these settings based on the setting of the *Status* field for the change request.

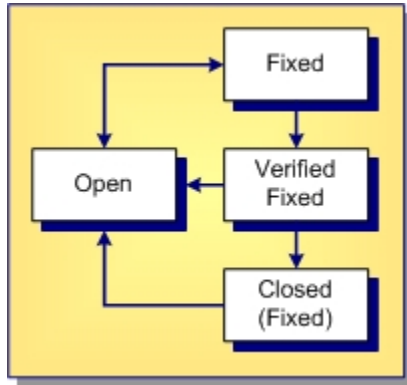
You cannot add additional settings to the *Status* field. However, you can rename them to better suit preferences set by your organization. For example, your organization may prefer to change the name of the Status *New* to *New Change Request*.

When you alter the status of a change request, the built-in workflow automatically selects the appropriate properties associated with the change in status.

After selecting *New*, *Open*, or *In Progress*, six new statuses display in the *Status* drop-down list box. These statuses, which are associated with the status you selected, are:

- ◆ *Deferred*
- ◆ *Cannot Reproduce*
- ◆ *As Designed*
- ◆ *Fixed*
- ◆ *Documented*
- ◆ *Is Duplicate*

Lifecycle for Change Requests



The above diagram shows the lifecycle for a change request with an initial status of *Open*. The status was then set to *Fixed*. After this setting, the built-in workflow added an additional status field of *Verified Fixed*. Finally, the change request was closed, meaning its status was set to *Closed (Fixed)*.

The diagram also shows that a change request can be reopened at any stage in its lifecycle because the arrows leading from each of the three fixed statuses can lead back to the *Open* status at any time.

Summary of the Change Request Automatic Workflow

The following table summarizes the steps used in processing change requests as explained in this topic. It includes the automatic workflow changes the application makes to change requests based on their statuses.

Change Request Process Management

Step	Description
Submit	<p>Anyone (usually a tester or quality assurance engineer) can submit a change request.</p> <p>Process: Select the Change Request tab. Then select New from the Change Request or context menu.</p> <p>A change request has the following default properties (which you can change if necessary).</p> <p>Status: New</p> <p>Severity: Low</p> <p>Priority: Not prioritized</p> <p>Type: Defect</p> <p>Platform: All</p> <p>Last Build Tested: Current build label</p> <p>Entered by: Person currently logged on to the application</p> <p>Many other fields are initially blank. Some team leaders prefer to have all change requests submitted at the root folder. They use drag-and-drop to move the change requests to the appropriate child folders.</p>
Assign*	<p>Process: The team leader finds all new change requests and does one of the following:</p> <ul style="list-style-type: none">■ Opens the change request and assigns it to a developer, help writer, or other appropriate team member.■ Defers the change request until a later date, perhaps the next release of the product.■ Specifies that the change request is <i>As Designed</i> and not to be fixed. If the change request status is <i>Open</i>, no automatic changes occur. If the change request status is <i>Deferred</i> or <i>As Designed</i>, then <i>Addressed in Build</i> is disabled and the responsibility is assigned to the user who created the change request.
Resolve	<p>Process: Users find the <i>Open</i> or <i>In Progress</i> change requests assigned to them, and do one of the following for each request:</p>

- Resolve the problem in the system and update the properties of the change request. (The statuses that indicate that a change request has been resolved are *Cannot Reproduce*, *As Designed*, *Fixed*, *Documented*, or *Is Duplicate*.)

- Defer the change request until a later date, perhaps the next release of the product. Your team leader may prefer that you do not defer change requests.

If the change request status is one of the possible resolution statuses, then *Addressed in Build* becomes *Next Build* for *Fixed* and *Documented* statuses. It becomes disabled for other statuses. By default, the responsibility is assigned to the person who submitted the change request, who is expected to verify the resolution.

If the change request status is *Deferred*, then *Addressed in Build* is disabled and the responsibility is assigned, by default, to the user who created the change request.

Build	<p>Who builds the project? The project view may have a formal or informal build process. However, at some point, all the files, and so on currently in the view receive that build label. It is usually applied to the source code files, and so on that were compiled (and may need to be changed) rather than to the executable files that result from the build.</p> <p>Effect on change requests: For any resolved change request that has <i>Next Build</i> as the setting for its <i>Addressed In Build</i> property, <i>Next Build</i> is replaced with the next build label that is created.</p> <p>Note: If a new build label is based on a past configuration (rather than the current configuration), it has no effect on the <i>Addressed In Build</i> property.</p> <p>If a change request has not branched in its current location, <i>Next Build</i> may be replaced with a build label from another view. For example, suppose you create a branching child view or share a folder from one view to another. Suppose that <i>Next Build</i> is the value of some change request's <i>Addressed In Build</i> property and that change request has not branched. When a build label is created in the source view, <i>Next Build</i> is replaced with the name of that build label, regardless of the location.</p>
Verify*	<p>The person who submitted the change request (usually a tester or quality assurance engineer) verifies a resolution.</p> <p>Process: Install the build in which the resolution is to be verified and determine whether the change request has been resolved correctly. Do one of the following:</p> <ul style="list-style-type: none">■ Verify the change request, marking it as <i>Verified Cannot Reproduce</i>, <i>Verified As Designed</i>, <i>Verified Fixed</i>, <i>Verified Documented</i>, or <i>Verified Is Duplicate</i>.■ Reopen the change request and update the setting for <i>Last Build Tested</i>. <p>If the change request status is <i>Verified</i>, no automatic changes occur.</p> <p>If the change request status is <i>Open</i>, <i>Addressed in Build</i> is blank. If the change request has changed from resolved to <i>Open</i>, the user who changed the status to <i>Fixed</i> or <i>Documented</i> becomes responsible.</p>
Close*	<p>Usually the team leader closes the change request.</p> <p>Process: The team leader does one of the following:</p> <p>Reviews and closes the verified change request.</p> <p>Reopens the change request.</p> <p>If the change request status is <i>Closed</i>, then no automatic changes occur.</p> <p>If the change request status is <i>Open</i>, then <i>Addressed in Build</i> is blank. If the change request has changed from resolved to <i>Open</i>, the user who changed the status to <i>Fixed</i> or <i>Documented</i> becomes responsible.</p> <p>If the status of a change request changes from <i>Verified</i> to <i>Open</i>, the user who changed the status to <i>Fixed</i> or <i>Documented</i> becomes responsible and <i>Addressed in Build</i> is blank.</p>

*Changes in status can result in automatic changes to other properties.

Related Procedures

[Working with Change Requests](#)

Queries

You can use a query to limit the items displayed in the upper pane. Each query is performed on all items in the StarTeam folder and component you have selected. The fields included in the query do not have to display in the upper pane. Once a query has been created, it can be used in every project in the same server configuration.

StarTeam queries have the following attributes:

- ◆ A unique name that easily identifies the query. Query names are not case-sensitive.
- ◆ Public or private status. Anyone with appropriate access rights can use public queries, while private queries are available only to your user ID. Once a query has been saved with a specific status, its status cannot be changed. However, you can copy a query and change the state of the new query.
- ◆ A logical expression appropriate for items of a particular type. These expressions include one or more conditions. A condition consists of a field (not necessarily a current column header), a relational operator, and a value to be compared to the value of the field. For example, a condition used to locate change requests might be: `Responsibility Equals Rhonda Thurman`. More complex queries include two or more conditions bound together by logical operators: AND, OR, and NOT. For example, to locate all the change requests for which Rhonda Thurman is responsible that also have a high severity, use: `Responsibility Equals Rhonda Thurman AND Severity Equals High`.

Note: If you are creating a complex query, and the first logical operator in your query should be **OR**, select the **AND** logical operator in the query tree; then click the **AND->OR->NOT** button. This changes an **AND** to an **OR**. Similarly, one more click changes the **OR** to a **NOT**. Keep toggling the button until the operator that appears is the one you want to use. It is best to use the condition or logical operation that will result in the fewest matches as the first condition or logical operation.

Related Procedures

[Creating Queries](#)

[Querying Data](#)

[Filtering Data](#)

Reports

StarTeam offers a wide variety of reports, which are pre-formatted in HTML. However, you can modify the HTML formatting or the columns used in the report by modifying the correct template for the report. You can also use StarTeam Datamart to extract data from StarTeam Server and place the data into a relational database where reporting tools (Crystal Reports and Business Objects) can access it.

Report generation is affected by sorting, grouping, and selecting items in the file, change request, topic, or task list of the view window. Before you generate a report, arrange the data in the upper pane of the client.

This topic provides some examples of why you might generate reports and information about customizing report templates.

Scenarios for Generating Reports

Reports are generated for a variety of reasons. The following examples describe three possible scenarios for generating a report using change requests and a brief walkthrough on how to generate them.

A manager wants a report of all the change requests that have been resolved across the entire project:

- 1 The manager selects the root folder from the folder hierarchy, and selects the Change Request tab. The list of change requests display in the upper pane.
- 2 The manager selects the *Status=Resolved* filter from the Status drop-down list box.
- 3 The **All Descendants** (either from the toolbar or the **Change Request** menu) button is activated.
- 4 Finally, he or she selects **Change Request ► Reports** from the main menu to generate a *Change Request Summary Report*.

A team leader prefers to see change requests based on who is responsible for fixing them:

- 1 The team leader selects the project folder that corresponds to his or her team from the folder hierarchy, and selects the **Change Request** tab. The list of change requests display in the upper pane.
- 2 The team leader selects the **<Show All>** filter from the Status drop-down list box.
- 3 The **All Descendants** button is activated.
- 4 The team leader clicks the header of the **Responsibility** column to sort the change requests based on the responsible team member.
- 5 Finally, the team leader selects **Change Request ► Reports** from the main menu to generate a *Change Request Detail Report*.

Individual programmers want to see only those change requests for which they are responsible:

- 1 The individual programmers select their project folders from the folder hierarchy, and select the Change Request tab. The list of change requests display in the upper pane.
- 2 The programmer applies a query (Responsibility Equals **<username>**) to view only his or her assigned change requests.
- 3 The **All Descendants** button is activated.
- 4 Finally, the programmer selects **Change Request ► Reports** from the main menu to generate a *Change Request Detail Report*.

Although these examples apply only to change requests, you can create additional reports for other items. You can also export data for use in spreadsheets and other applications.

Report Templates

StarTeam allows you to customize report templates. The templates are located in the folder you designated during installation. For example, if you used the default installation path for the Cross-Platform Client on a Windows platform, the Reports folder is `C:\Program Files\Borland\StarTeam Cross-Platform Client 2006\Reports`. Be aware that different clients and different releases of just one client will probably have different installation folders. You may need to put templates in more than one location. All the templates are in HTML format.

You can open and edit the report templates in any text editor, Microsoft's Developer Studio, or HTML tool. A simple and easy method of creating and editing templates is to use Microsoft Word 97 or later, which includes automated HTML file generation.

The Reports folder includes a series of templates for each type of report. Each template provides the formatting information needed to create a part of the report. For example, the *Change Request Default* report uses the following templates:

- ◆ ChangeDefault.Title
- ◆ ChangeDefault.GrpInfo
- ◆ ChangeDefault.Group1
- ◆ ChangeDefault.EndReport

The xDefault.Title is processed first and only once. It uses the title you provide in the Reports dialog as the Report Title.

The xDefault.GrpInfo is processed once for each group. Although you can sort data by clicking column headers, this does not result in groups for a report. You must use the Sort and Group feature to arrange groups.

The xDefault.Group1 template is repeated for each item, in this case, each change request, in the report. It creates a record for the item and separates it from the records for other items with a horizontal line.

The xDefault.EndReport is processed only once. It totals the number of items in the report.

Refer to the link at the bottom of this topic for important reference information about the contents of the report templates.

Related Procedures

[Creating Reports and Exporting Data](#)

[Generating Reports from a File Compare/Merge Session](#)

[Viewing and Printing a View Compare/Merge Difference Report](#)

Related Reference

[Reports](#)

Charts

The Cross-Platform Client and StarTeam Visual Studio Integration enable you to create simple, distribution, correlation and time-series charts of your data.

The charts you can create depend upon the component tab menu option that you have selected. Thus, the charts available for files differ from those available for change requests.

All application charts are based on the data shown on the upper pane in the project window. To select the data to be used for a chart, you can show or hide all descendants of a folder, sort and group items, and run queries and filters. You can also display additional fields, if desired.

You can display a maximum of 60 fields in the upper pane.

Related Procedures

[Creating Charts](#)

Related Reference

[Charts](#)

Filters

A filter is a named arrangement of data that consists of a set of fields (used as column headers), sorting and grouping information, and (usually) a query. Once a filter has been created, it can be used in every project that has the same server configuration.

Filter names are not case sensitive. For example, if you have a filter named *recent CRs*, you cannot create a filter named *Recent CRs*, as StarTeam considers the two filters to be identical. In the **Filters** list box, filters display in alphanumeric order, but you can control the order in which they appear by carefully naming or renaming them.

If you set up a filter and do a **Send to** in the client, only the fields displayed by the filter are sent to the recipient.

You can filter data in the upper pane in several different ways:

- ◆ By applying an existing filter.
- ◆ By arranging the data (changing displayed fields, sorting and grouping the files, and so on) and applying a query. You can then use this arrangement as the basis for a new filter.
- ◆ By creating a new filter from scratch.

Note: Only private queries can be used in private filters, and only public queries can be used in public filters. Therefore, you cannot copy a filter and change the status of the new filter unless the filter does not include a query.

Related Procedures

[Querying Data](#)

[Filtering Data](#)

Related Reference

[Filters Reference Topics](#)

Fields

StarTeam provides both common and advanced fields that you can display as columns in the upper pane or use in queries along with relational operators that you can use with those fields to define conditions. To see an advanced field in the **Show Fields**, **Sort and Group**, or **Query** dialog boxes, you must select the **Show Advanced Fields** check box.

The related reference topic at the end of this topic contains field descriptions including the internal identifier for the field. In report templates, you must use the internal identifier, instead of the field name. Most internal identifiers contain no spaces. However, some do have spaces. Internal identifiers are case sensitive. You can also use these fields when defining customized Email notification templates and custom HTML-based templates that modify the Detail panes.

Note: Client-calculated fields cannot be used in custom email notifications or with Notification Agent.

Related Procedures

[Querying Data](#)

[Filtering Data](#)

[Creating Reports and Exporting Data](#)

[Customizing the Detail Pane](#)

Related Reference

[Fields](#)

Data Export with Datamart

StarTeam Datamart is a complementary product to StarTeam Server. StarTeam Datamart uses the StarTeam SDK to communicate with StarTeam Server to create a reporting database that you can use with popular third party reporting applications such as Crystal Reports and Business Objects (reporting applications are not included with StarTeam Datamart).

StarTeam Datamart extracts data from a StarTeam server and places the data into a relational database, where reporting tools can access it. The data is stored in meaningful text instead of IDs.

StarTeam Datamart can extract information from every project, every view in each project, every folder in each view, and every item in each folder, as well as labels, links, and history for each item. You can restrict the extracting to a particular project and view, and also restrict to extract only for certain tables.

A sample Business Objects universe and sample reports in both Business Objects and Crystal Reports formats are included. These samples can be used as a basis for creating custom user defined reports.

- ◆ **Datamart Extractor:** extracts the data from a StarTeam Server and puts the data into a relational database. For StarTeam Administrator use.
- ◆ **Datamart Synchronizer:** opens the universe and synchronizes the data to reflect any field changes. For StarTeam Administrator use.

Note: The Synchronizer and Viewer work with Business Objects 6.1 universes and reports. The Viewer also works with Crystal Reports 10 reports. You can view StarTeam Datamart reports with the Crystal Reports Viewer: [http:// www.businessobjects.com/products/reporting/crystalreports/eval.asp](http://www.businessobjects.com/products/reporting/crystalreports/eval.asp)

Cross-Server Configuration/Project Searches with Borland Search

Borland Search allows users to perform ad hoc queries across projects. The query results reflect the access rights of the user logged on to Borland Search so information is shared across the organization without compromising security.

Borland Search includes an Administrative Console for controlling the data sources, extractions, and indexing that provides searchable data to the end users. This console provides real-time status information as well as monitoring and configuration tools.

For more information, download the documentation from the following web site: <http://info.borland.com/techpubs/starteam>.

Tour of the UI

This section contains conceptual topics describing the StarTeam user interface.

In This Section

[Cross-Platform Client](#)

This section contains UI overview topics related to the Cross-Platform Client.

Cross-Platform Client

This section contains UI overview topics related to the Cross-Platform Client.

In This Section

[Cross-Platform Client Overview](#)

This topic describes the UI of the Cross-Platform Client.

[Change Perspective UI](#)

This topic describes the UI for the Change Perspective in the Cross-Platform Client

[Chart Window](#)

This topic describes the UI for the Chart window.

[View Compare/Merge \(VCM\) UI](#)

This section contains overview topics related to the View Compare/Merge UI.

[File Compare/Merge UI](#)

This topic describes the UI for File Compare/Merge (FCM.)

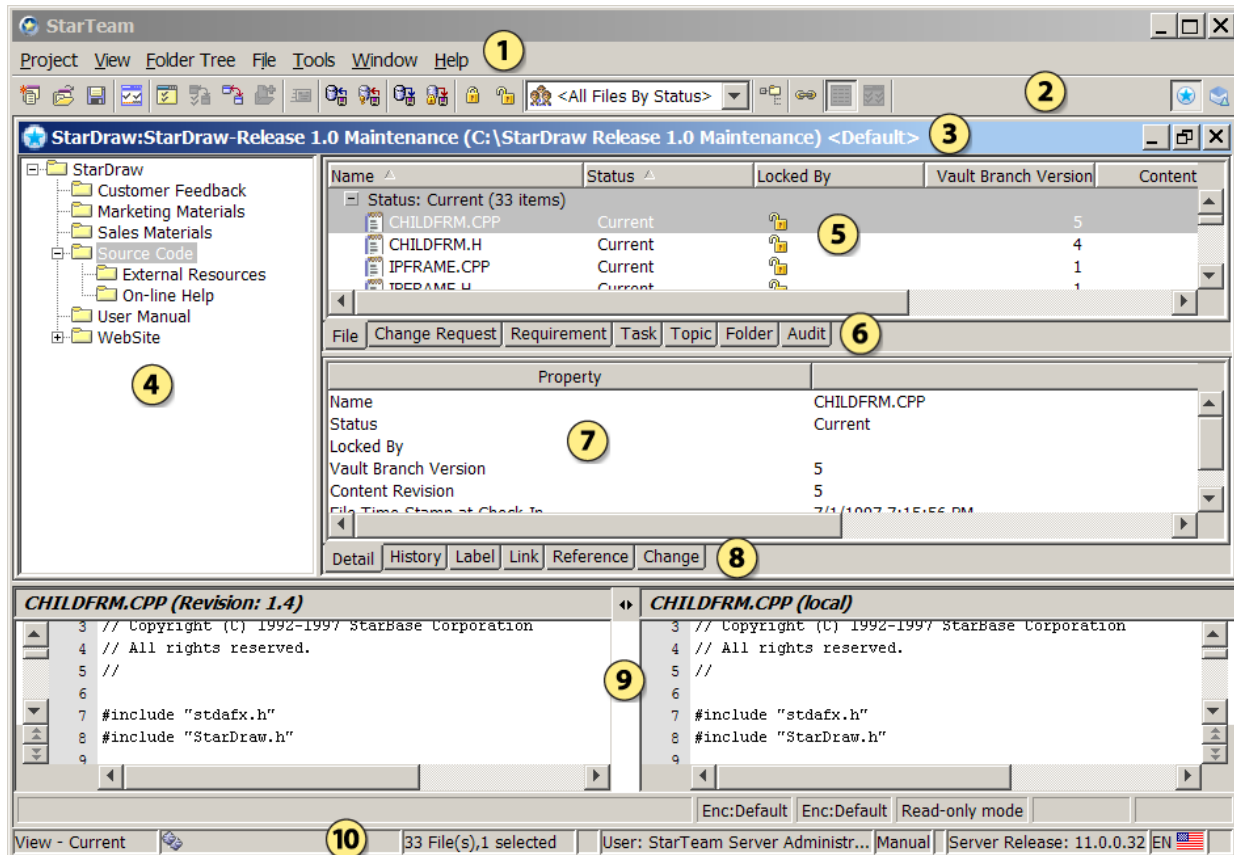
[Toolbar Utility](#)

This topic describes the UI for the Toolbar Utility.

Cross-Platform Client Overview

This topic describes the UI of the Cross-Platform Client.

Cross-Platform Client UI



- | | | |
|----------------|--------------------|--------------------------|
| 1 Main Menu | 5 Upper Pane | 9 Embedded Compare Panes |
| 2 Main Toolbar | 6 Component Tabs | 10 Status Bar |
| 3 Title Bar | 7 Lower Pane | |
| 4 Folder Tree | 8 Information Tabs | |

This window is also called the project view window, because it shows not only a project but a particular view of the project. That view may be the root (or initial) view with which the project was created. The rest of this topic describes the numbered components in the above image.



Main Menu

The contents of the main menu vary depending upon the selected component. In this example, the File component has focus, so the **File** menu displays as a selection in the main menu. Possible components include File, Folder, Change Request, Requirement, Task, Topic, and Audit. Menu items that always display in the main menu, regardless of the selected component, include **Project**, **View**, **Folder Tree**, **Tools**, **Window**, and **Help**.

Toolbar

The buttons on the toolbar differ according to the selected component. Frequently used main menu commands have corresponding buttons on the toolbar. Fly-over text displays when you hover your mouse over the toolbar buttons.

At the right of the toolbar are two buttons for switching between two perspective windows of the StarTeam information:

- ◆  : The standard **Content Perspective** which is the perspective you are accustomed to using for Cross-Platform Client.
- ◆  : The **Change Perspective** which allows you to view and manage all your View Compare/Merge change packages from past VCM sessions.

Title Bar

The title bar at the top of the main window displays the following information:

- ◆ Server configuration that contains the currently displayed project view
- ◆ Project name
- ◆ View name
- ◆ Working folder for the view

Folder Tree


The left pane of the main window displays the folder tree or folder hierarchy. These folders have a one-to-many relationship with the contents of the upper pane because each folder can contain different types of items (for example, files and change requests). Using the Folder Tree in the left pane, you can perform the following operations:

Display information using icons	A folder icon precedes each folder name. Clicking this icon displays folder-associated information in the upper pane. The information displayed depends on the folder and the tab selected on the upper pane.
Expand and collapse branches	<p>You can expand or collapse branches of the folder tree. A plus sign identifies a collapsed branch. Clicking it expands the branch. A minus sign identifies an expanded branch. Clicking it collapses the branch. Alternatively, you can double-click folder names to expand or collapse branches of the folder tree.</p> <p>You can also move through the folder tree using the UP ARROW, DOWN ARROW, LEFT ARROW, and RIGHT ARROW keys. The RIGHT ARROW expands a folder. The LEFT ARROW collapses a folder.</p>
Right-click for context menu	After selecting a folder from the folder tree, right-click to display a context menu that presents the same choices as the Folder Tree menu.
Drag items and folders	<p>To move a child folder, you can click on it and drag it from one parent folder to another. You can also drag items from the upper pane to the tree in the left pane for the current view or another view (if both are in the same server configuration).</p> <p>To move items from one folder or view to another, select the items, and then drag them from one location to the other.</p>

Upper Pane

The upper pane of the main window contains a tree or list of items associated with the folder that displays in the folder tree. Although each folder can contain items of different types, the upper pane displays only one type of data at a time. The type of data displayed depends upon the component tab selected.

The contents displayed in the upper pane depend upon the following factors:

- ◆ The folder selected from the folder tree in the left pane.
- ◆ The component tab selected from the upper pane.
- ◆ The filter selected from the **Filter drop-down list**, which displays in the toolbar above the upper pane.
- ◆ The **All Descendants**  toolbar button, found in the toolbar above the upper pane. The **All Descendants** command determines the depth for which the client displays information. When not selected, the client displays information for the selected folder only. When selected, the client displays information for the selected folder, its children, its children's children, and so on.

Component Tabs

The upper pane of the main window contains component tabs that represent the components that your StarTeam license provides. Clicking on one of the component tabs in the upper pane brings that component tab into focus.


StarTeam Enterprise Advantage licenses provide all components:

- ◆ File
- ◆ Change Request
- ◆ Requirement
- ◆ Task
- ◆ Topic
- ◆ Folder
- ◆ Audit

Lower Pane and Information Tabs

The lower pane of the main window displays information about the item selected on the upper pane. The five information tabs on the lower pane show different types of information. All of the tabs apply to all of the upper pane components except the **Audit** component. When you select the **Audit** component, you see only the **Detail tab**. Each of the information tabs are described below.

Detail tab	<p>The Detail tab lists the information displayed on the upper pane in a horizontal format. For example, if you select a file from the upper pane, the lower pane will show its name, size, status, and so on, depending on the columns that appear in the upper pane.</p> <p>You can modify the display format and content of the Detail (lower) panes in the Cross-Platform client on a per-workstation basis. For more information, refer to the link “Customizing the Detail Panes in the Cross-Platform Client” at the end of this topic.</p>
History tab	<p>The History tab lists the past revisions for the selected item. The only exception is a file with the <i>Not In View</i> status, in which case the tab displays nothing at all.</p>
Label tab	<p>The Label tab lists the labels associated with each revision of the selected item. You can use drag-and-drop to move a label from one revision to another.</p>

Link tab	The Link tab shows the items to which the selected item is linked. For example, a file might be linked to a folder, another file, or a change request. Items can also be linked externally from one server to another. The Link tab decorates the external link in the View column with a special external link decoration (), and provides a checkbox which you can use to show or not show external links.
Reference tab	The Reference tab shows the relationships between the selected item and other folders or items with which it is associated. A folder or item may be associated with more than one project, view, or parent folder in the same server configuration because of sharing or because a child view has been created. Each instance of the original folder or item has a reference. Item references can be viewed on the Reference tab of the lower pane. To see folder references, you must select Advanced ▶ References from the Folder or context menu to display a dialog.
Change tab	The Change tab shows all the changed that have been made to a change package when in the Change Perspective .

Embedded File Compare/Merge

The File Compare/Merge tool is also embedded in the client and when activated, is available for use with the **File**, **History**, and **Link** (if the link is to a file) tabs. You can use this tool in its embedded format to show the differences in the file that you have selected in the upper pane and your local file.

By default, the client does not display the embedded version of File Compare/Merge. You can toggle this feature on and off using the **Tools ▶ File Compare** main menu command.

Note: File Compare/Merge also opens in standalone mode. You can open File Compare/Merge using the start menu.

Status Bar

By looking at the status bar at the bottom of the main window, you can determine the following:

- ◆ Whether a view is current, rolled back to a label, rolled back to a promotion state, or rolled back to a specific time. If a view is read-only, *R/O* displays before the configuration information. All rolled back views are read-only because you cannot change the past.
- ◆ Whether you have selected an active process item. For example, if you have select a change request for your active process item, the status bar displays the change request title along with its associated number.
- ◆ The number of items in the upper pane and the number of items selected.
- ◆ Whether you have applied a query to the upper pane.
- ◆ Name of the logged-on user.
- ◆ The status of StarTeamMPX. A lightning bolt icon represents the status of StarTeamMPX. The lightning bolt changes color or shows a small red "disconnected" symbol in the lower left corner to let you know if StarTeamMPX is available, available but disabled, or available and disconnected. The words **Instant**, **Auto**, and **Manual** provide refresh status information.
- ◆ The version of StarTeam Server that the client is accessing.
- ◆ The locale and default language of the client.
- ◆ Whether you have new (or unread) change requests, requirements, tasks, or topics assigned to you.

Related Concepts

[References to Folders and Items](#)
[Where to Find Documentation for Each Product](#)
[StarTeam Product Overview](#)
[Change Perspective UI](#)

Related Procedures

[Working with Folders and Items](#)
[Sorting and Grouping Data](#)
[Refreshing Views](#)
[Customizing the Detail Pane](#)
[Comparing and Merging Files and Folders](#)
[Linking and Unlinking Items](#)
[Managing Labels and Promotion States](#)


Change Perspective UI

This topic describes the UI for the **Change Perspective** in the Cross-Platform Client. The **Change Perspective** gives you a view of all the View Compare/Merge change packages which have been created, and either are committed or are still uncommitted. The **Change Perspective** lets you manage change perspectives, review all the information about change packages, review what changes have been made to in each change package, or what changes are being proposed for an uncommitted change package. Managers can review change packages before they are committed, and if appropriate, make additional changes and commit the sessions themselves.

All the changes in a committed change package can be *Replayed* (applied to another view) using the menu drag and drop, saving the effort and time of re-creating from scratch the same changes in a new VCM change session for another view.

Note: The **Change Perspective** is actually an alternate view window in the main Cross-Platform Client which can

be displayed by clicking on the **Show Change Perspective** icon  on the right side of the toolbar.

Clicking the **Show Content Perspective**  icon switches you back to the familiar Cross-Platform Client. Each perspective has its own menu and set of toolbar buttons.

Change Perspective Layouts

When you switch to the **Change Perspective**, you see the default “side by side” layout with the changes packages listed on the left, and the **Details** pane on the right. If you have performed and saved any View Compare/Merge sessions prior to showing the **Change Perspective**, you will see them in the change perspectives list. Which change packages are displayed depends on the filter you select at the top of the list.

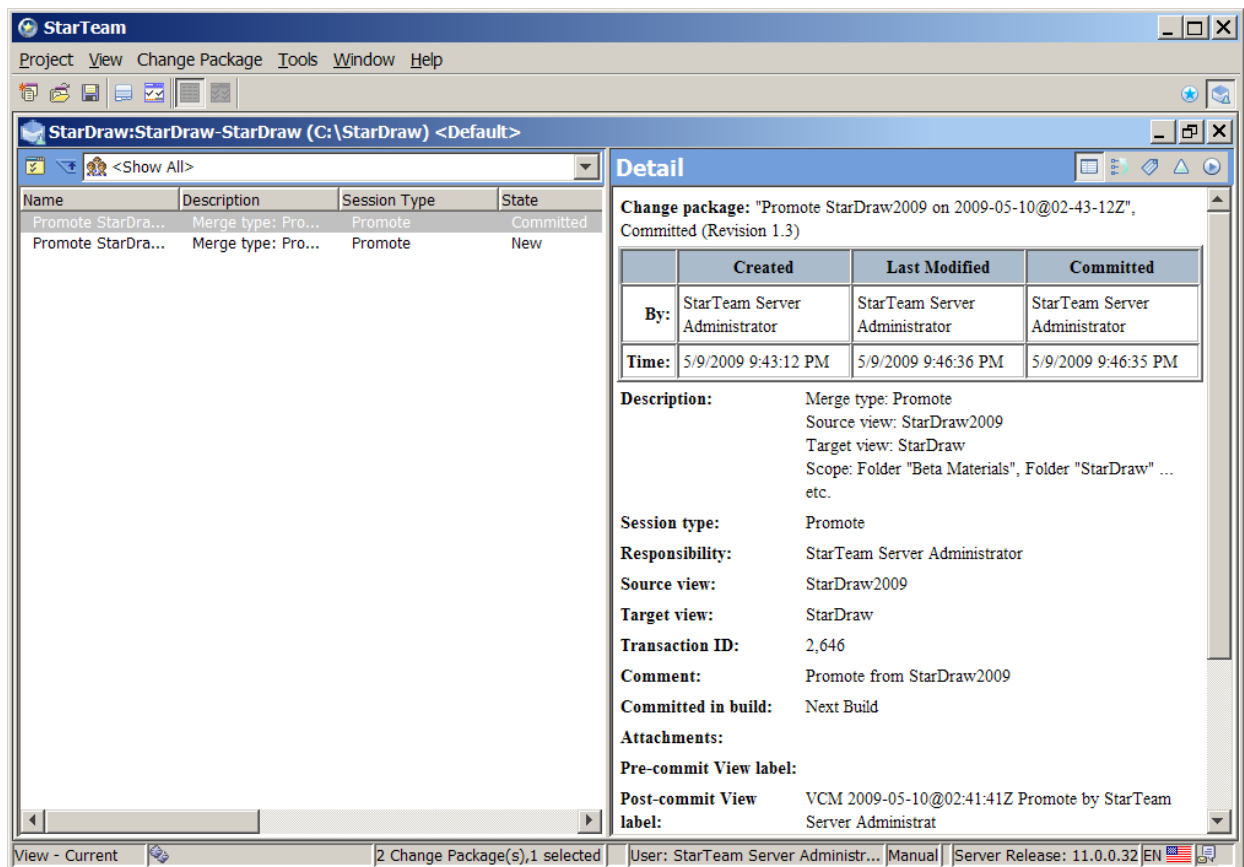
StarTeam gives you the option of using one of the two following layouts for the Change Perspective: the “side by side” vertical arrangement, or the “top over bottom” horizontal arrangement.

Side by Side Layout

The “side by side” layout lists the change packages in a column on the left, and the **Detail**, **History**, **Label**, **Change**, and **Replay** views in a pane on the right.

The **Change Perspective** uses some of the regular views from the **Content Perspective**, but it also has two different views for Change Perspective and Change actions in particular: The **Change**, and **Replay** views. These are discussed in more detail later.

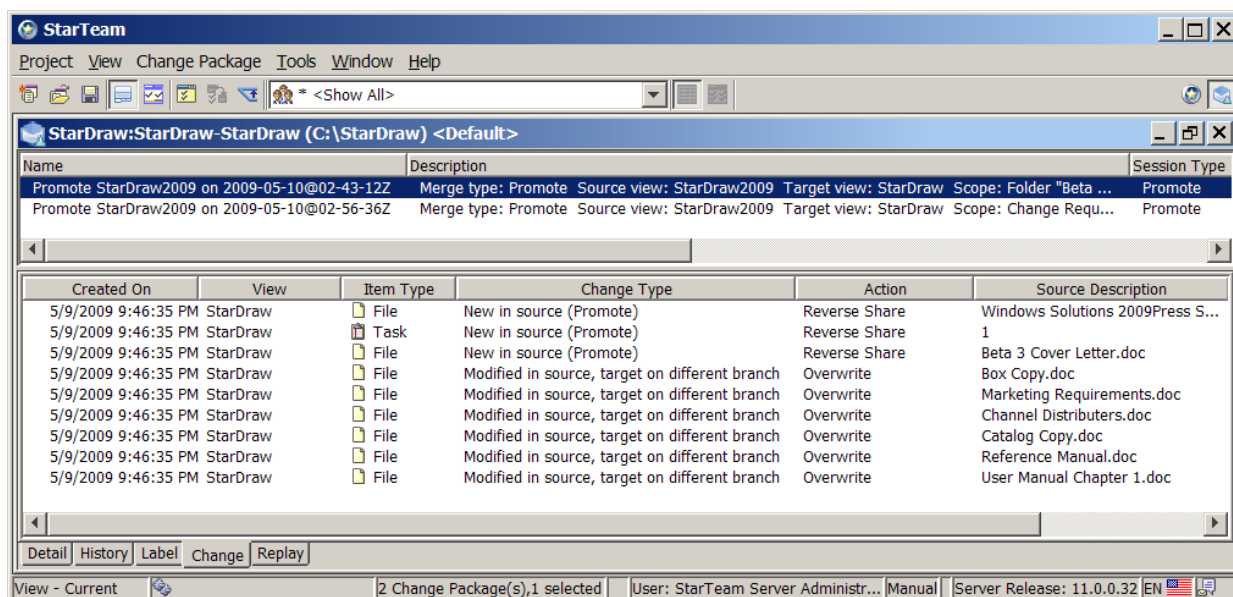
In the “side by side” layout, clicking a toolbar button at the top right of the view pane switches the view. Those view toolbar buttons correspond to the tabs in the “top over bottom” view, and have tool tips to show you their functions.



Top over Bottom Layout

The “top over bottom” layout shows the change packages in the top half of the window, and the **Detail**, **History**, **Label**, **Change**, and **Replay** views in tabbed panes at the bottom, much like the typical Cross-Platform Client **Content Perspective**.

Note: To switch to the “top over bottom” layout, click the **Layout Top Over Bottom** button  on the main toolbar.



Change Perspective Toolbars, Tabs, and Menus




Each **Change Perspective** layout uses either toolbars, tabs, and menus to perform the different change package tasks, or to see different views. As with StarTeam in general, each toolbar button has a corresponding menu item, either on the **Change Package** menu, or on the context menu inside the **Change Perspective**.

Toolbars and Tabs

The “side by side” **Change Perspective** layout uses toolbars at the top of each side to perform functions or change views.

Each side in this layout has its own toolbar. On the left, above the list of change sessions, there are two icons and the filter drop-down list.


Side by Side Layout Toolbar on Left


Item	Description
	Properties: Displays the Change Properties dialog box for the selected change package. You can view the change package properties, or if you have the proper access rights, you can modify the properties.
	Open (in VCM): Opens the selected change package in a VCM session. If the change package is new and uncommitted, you can resolve conflicts and commit the session. However, if the selected change package has already been committed, it opens in the VCM session in Read-Only mode. No changes can be made once the session is committed.
 <All Change Packages By State & Responsibili...	Filter drop-down box: Displays the list of change package sessions according to the selected filters. As with the rest of StarTeam, you can create new filters and queries for the Change Perspective .


Note: The toolbar items on the left move up to the main Cross-Platform Client window in the “top over bottom” layout of the **Change Perspective**.


Side by Side Layout Toolbar on Right


Item Description

- 
Detail View: Displays all the details for the selected change package in the VCM session in which it was created.
 A [changepackage.details.html](#) file is installed by default and utilized by the **Detail Viewer** which is fully customizable. See the procedure “Customizing the Detail Pane” in the documentation.

 - 
History View: Displays the historical information for the selected change package.

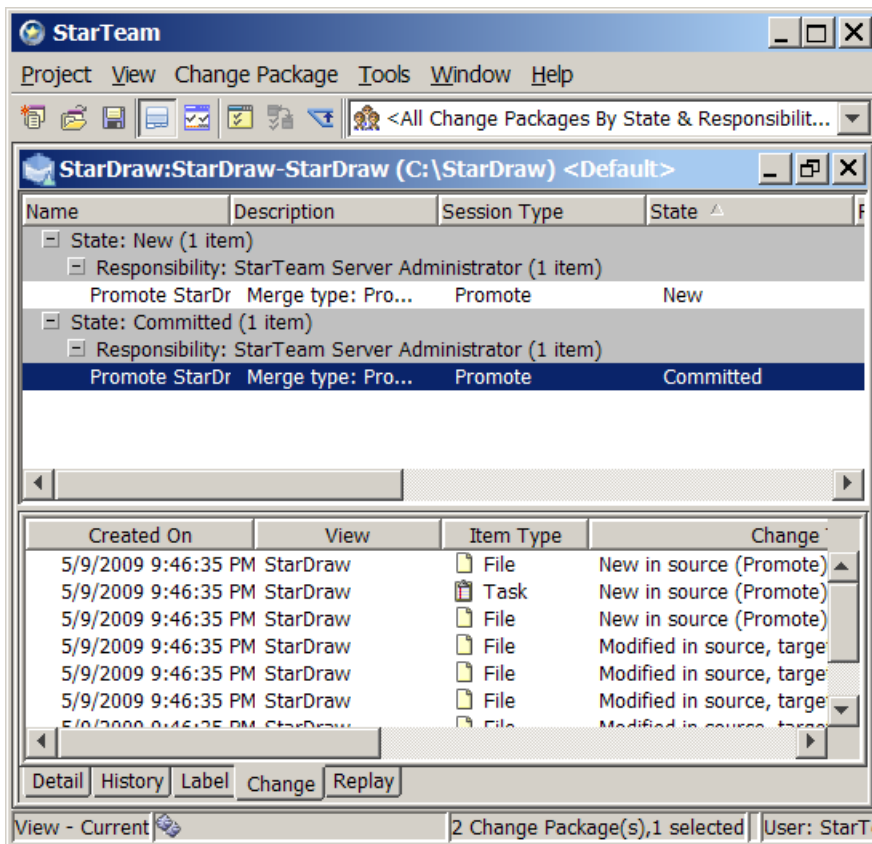
 - 
Labels View: Displays the branch and revision labels for the selected change package.

 - 
Changes View: Displays all the changes made in the selected committed change package.
 You can double-click on either a new or committed change package to see those changes in VCM session. You can continue working on the change package in VCM if it has not been committed, however, a committed change package will be read-only and cannot be changed.

 - 
Replay View: Displays all the possible views to which a change package can be replayed (the same changes reapplied to another view). simply by dragging the selected change package to the desired view in the Replay list. If the Replay is allowed, the icon displays a plus with it. If the Replay to the target view is not allowed, the cursor changes to a circle with a diagonal line.
-

Notice that when you switch to the “top over bottom” layout ,

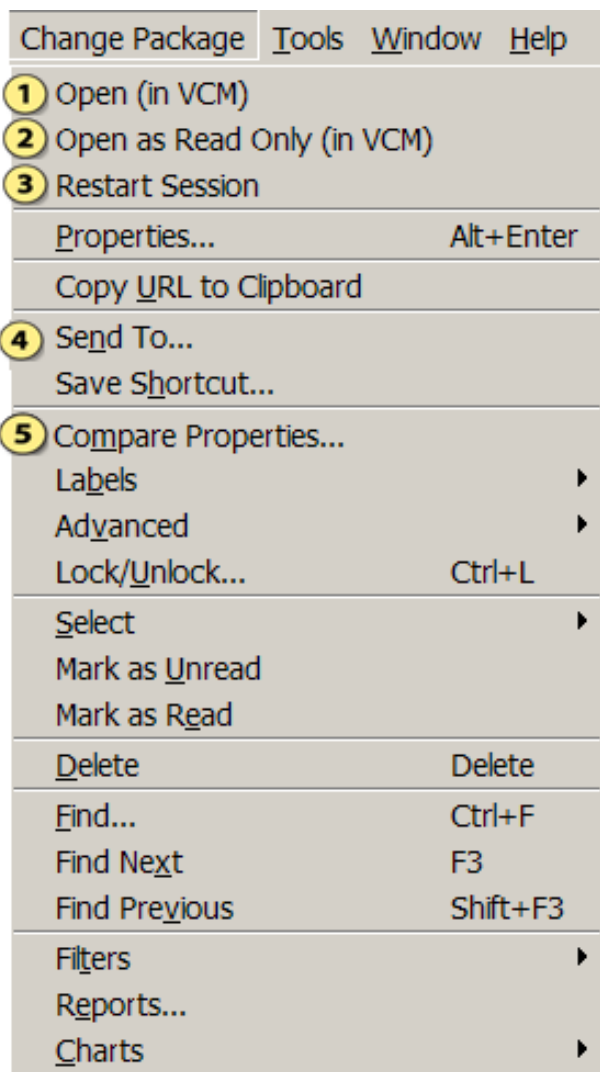
- ◆ The toolbar icons and **Filter** drop-down box move up to the main toolbar.
- ◆ The toolbar buttons for the different views that were on the right pane of the “side by side” view correspond exactly to the tabbed panes in the lower half of the “top over bottom” layout.



They perform exactly the same functions in either layout.

Change Package Menu

Most of the items on the **Change Package** menu are familiar to you from other item menus. However there are a few new menu items which are specific to change packages. These are numbered in the image below:

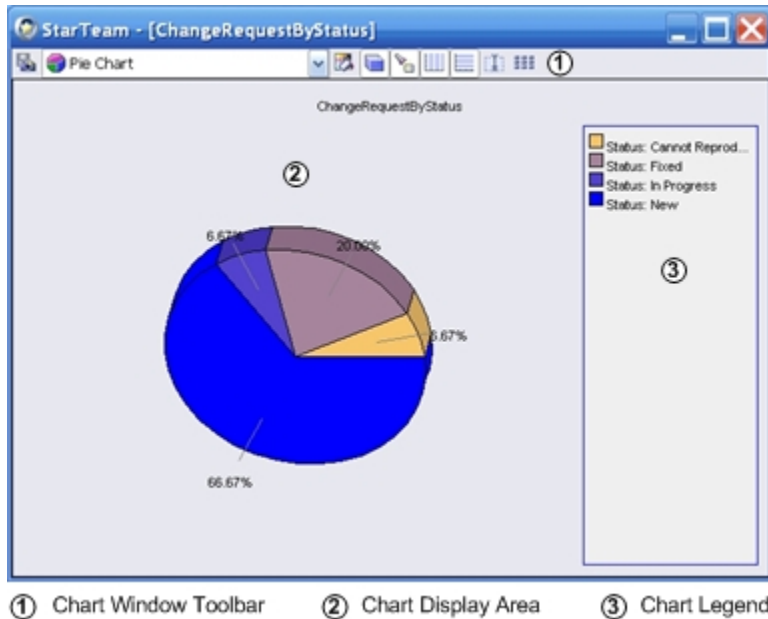


- 1 Opens a change package in a VCM Session in which changes can be made.
- 2 Opens a change package in read-only mode in a VCM session.
- 3 Restarts the change package in a VCM session at the point of just finishing the VCM Wizard so you can start over making all your changes.
- 4 When Send To is enabled on the server, you can email a change package.
- 5 Compares the properties of two selected change packages in the embedded editor compare panes which open at the bottom of the window.

Chart Window

This topic describes the UI for the Chart window.

Chart Window UI



The Cross-Platform Client and the StarTeam Visual Studio Integration enable you to generate charts from items in the upper pane.

The remainder of this topic discusses the numbered components in the above image.

Chart Window Toolbar

After generating a chart (see the “Creating Charts” link at the end of this topic for more information) the Chart Window opens displaying the chart of the items specified. The chart window toolbar enables you to :

- ◆ Save charts as .jpg files.
- ◆ Select a different chart type, such as an area chart or bar chart. For a list of chart types available, see the “Charts” link at the end of this topic.
- ◆ Edit chart colors.
- ◆ View charts in 2–D or 3–D.
- ◆ Optionally show chart legends or horizontal or vertical grid lines.
- ◆ Edit chart titles.
- ◆ Optionally show chart data in a table format.

Chart Display Area

The chart display area presents your chart data. You can select a different chart type using the drop-down list provided in the chart window toolbar.

Chart Legend

The chart legend displays by default. You can optionally display the chart legend (toggle on or off) using the **Toggle legend** toolbar button in the chart window toolbar.

Related Concepts

[Where to Find Documentation for Each Product](#)
[StarTeam Product Overview](#)

Related Procedures

[Creating Charts](#)

Related Reference

[Charts](#)

View Compare/Merge (VCM) UI

This section contains overview topics related to View Compare/Merge (VCM) UI. View Compare/Merge is the tool you use to compare two views or subsets of two views, referred to as the source and the target. As part of the process, you can merge corresponding items and make new revisions of those items in the target. These changes are contained in a Change Package object which can be saved uncommitted for review by another user, or which can be replayed (re-applied) to other views.

View Compare/Merge is available in the StarTeam Cross-Platform Client and the VCMUtility.

In This Section

[View Compare/Merge](#)

This topic describes the UI for View Compare/Merge (VCM.)

[View Compare/Merge Status Icons](#)

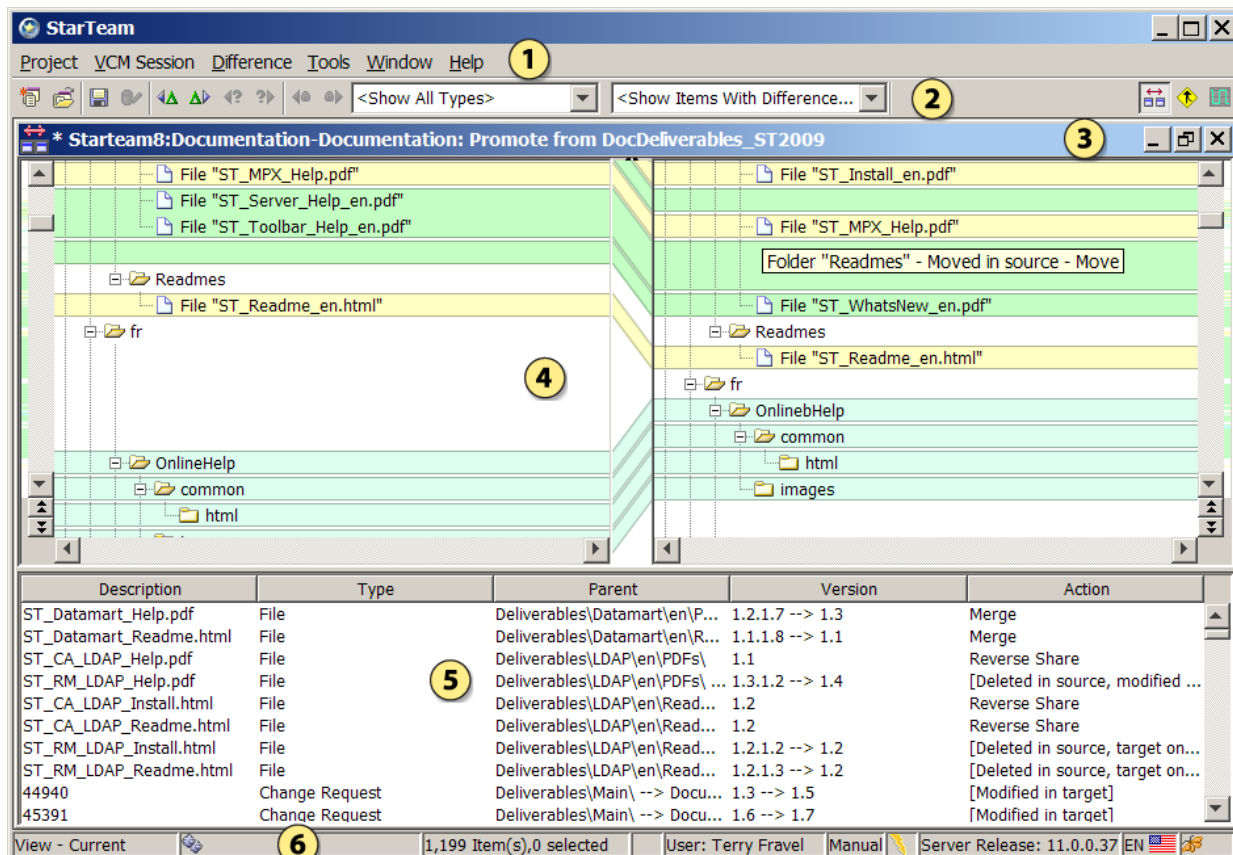
Table of View Compare/Merge icons.

View Compare/Merge

View Compare/Merge (VCM) is the tool you use to compare two views or subsets of two views, referred to as the source and the target. As part of the process, you can merge item differences in the source view to the target view. This topic describes the StarTeam View Compare/Merge UI.

Note: The View Compare/Merge UI is available in the StarTeam Cross-Platform Client and opens after running the **View Compare/Merge Wizard**.

View Compare/Merge UI



1 VCM Menu

3 VCM Title Bar

5 Lower Detail Pane

2 VCM Toolbar

4 Upper Comparison Pane

6 Status Bar

The VCM window provides three perspectives for comparing and merging views, and for resolving item differences.

- ◆ The **Compare Perspective**
- ◆ The **Merge Perspective**
- ◆ The **Test Perspective**

The image above displays the **Compare Perspective**. For more information on each perspective, see "View Compare/Merge Perspectives" in the Related Concepts links below.

VCM Menu

The VCM menu consists of a subset of the main client menus, plus two additional menus specific to View Compare/Merge:

- ◆ **VCM Session**, which enables you to view a VCM session summary, change perspectives, save, export, or delete your session, display a session report, and commit your changes.
- ◆ **Difference**, which enables you to view item properties, navigate to items with differences, merge file contents, and specify the merge action for an item.

VCM Toolbar

Frequently used VCM menu commands corresponding to the **VCM Session** and **Difference** menus have corresponding buttons on the toolbar. Tool tips display when you hover your mouse over the toolbar buttons. Filters specific to View Compare/Merge and the current perspective display in the **Filters** drop-down list on the **VCM Toolbar**.

Note: The **Refresh** button on the toolbar in the Merge Perspective does not change the snapshot of the source/target in use. It only refreshes the display with the View Compare/Merge session data that was already gathered. For instance, it might re-group items based on changes made by the user in the VCM Window. For example, if you go back to the target view and move a file, when you return to the merge window and selected **Window ▶ Refresh**, the moved file will still be shown in its original location.

Upper Pane

The upper pane serves a different function in each perspective:

- ◆ In the **Compare Perspective**, the upper pane displays two trees comparing the items in the source and target views. The tree nodes are highlighted to show the differences for all item types in the views or folders being compared.
- ◆ In the **Merge Perspective**, the upper pane displays the individual items on component tabs. The component tabs display VCM icons indicating there are resolved or unresolved conflicts of that item type in the selected folder. Icons also display in the **Merge Status** property column for each item with resolved or unresolved differences. See the topic "VCM Icons" in the links below for an image and description of each icon.
- ◆ In the **Test Perspective**, the upper pane shows the target check-out displayed in the same way as the **Merge Perspective**.

Lower Pane

The lower pane serves a different function for each perspective:

- ◆ In the **Compare Perspective**, the lower pane shows the details and merge action for the items in the VCM session. You can filter which items display in this perspective, and you can change the merge action for items in this pane.
- ◆ In the **Merge Perspective**, the lower pane contains three tabs. An item **Property** tab displays the properties from the source and target views for comparison, and enables you to edit them. Two **File Comparison** tabs compare the contents of the source and target files, or the target and merged files.
- ◆ The **Test Perspective** contains the same information tabs as the main client window.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing Selected Items in the Source and Target View](#)

Related Reference

[View Compare/Merge Status Icons](#)





[View Compare/Merge Wizard](#)

View Compare/Merge Status Icons

The status of folders and items in a View Compare/Merge session is represented in the **Merge Perspective** by icons which are explained in the tables below. Icons appear on or beside folders, items, and on component tabs. Icons change dynamically when conflicts are resolved.





When you hover the mouse cursor over an icon in the upper pane, a flyout menu appears from which you can choose a merge action.

Tooltips are available for folder icons.

Folder Status Icon	Meaning
	This folder contains compared items that remain unchanged.
	This folder contains compared items, some of which have unresolved changes called merge conflicts. Merge conflicts are differences in file contents or property values that require manual intervention.
	This folder contains compared items, none of which have unresolved changes. No manual intervention is necessary.
	This folder contains no compared items or folders.

The folder icons are displayed in the left pane beside each folder in the hierarchy tree of a View Compare/Merge session in the **Merge Perspective**. The folder icons in a View Compare/Merge session represent what is in the currently selected component tab in the upper pane. For example, in the situation where you have resolved all the file conflicts in a particular folder, but you still have unresolved change requests in that folder, the following will happen:

- ◆ When you click the **Files** tab, the folder icon will have a check mark, indicating that all file conflicts in that folder have been resolved.
- ◆ When you click the **Change Request** tab, the folder icon will have a question mark, indicating that the folder has unresolved change requests.

Item Status Icon	Meaning
No icon	Item is included in the comparison and is unchanged.
	Item resolved by View Compare/Merge. You can override the default action with a different action.
	Item has differences and is unresolved. You must resolve the differences manually.
	Item is not included in the comparison.
	Item is changed. This is used in Compare-only.

Item status icons are displayed in the upper pane of the **Merge Perspective** for each item with differences or conflicts. They are also displayed on each component tab that contains items with differences. Item icons appear in the **Merge Status** and **Merge Action** columns, and on individual property columns when properties are in conflict.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[View Compare/Merge Actions](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

[Merging Changes from a Child View to a Parent View \(Promote\)](#)

[Merging Changes from a Parent View to a Child View \(Rebase\)](#)

[Merging Changes Between Any Two Related Views \(Replicate\)](#)

[Resolving Merge Conflicts in View Compare/Merge](#)

[Testing View Compare/Merge Changes](#)

[Saving a VCM Session to a Change Package](#)

[Changing View Compare/Merge Actions](#)

File Compare/Merge UI

This topic describes the UI for File Compare/Merge (FCM).

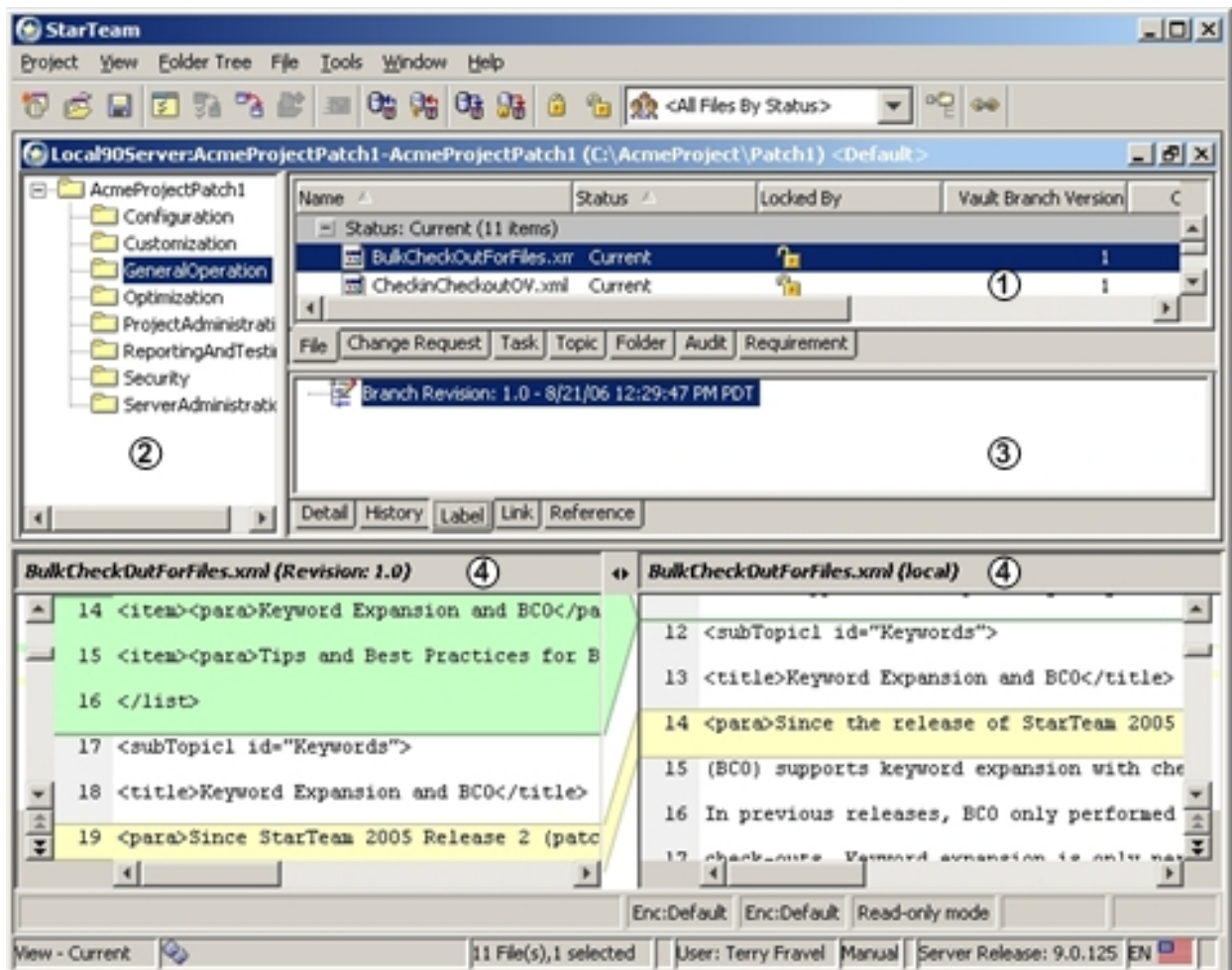
There are three versions of File Compare/Merge:

- ◆ The embedded File Compare/Merge, where the FCM panes are embedded at the bottom of the StarTeam client window. These panes are for comparison only.
- ◆ In a separate pop-up window inside the StarTeam client where you can compare and merge local files and repository files.
- ◆ In a standalone File Compare/Merge window outside the StarTeam client where you can compare and merge local files and folders, and compare and move images.

Each version of File Compare/Merge provides different features and functions and is explained briefly below:

Embedded File Compare/Merge

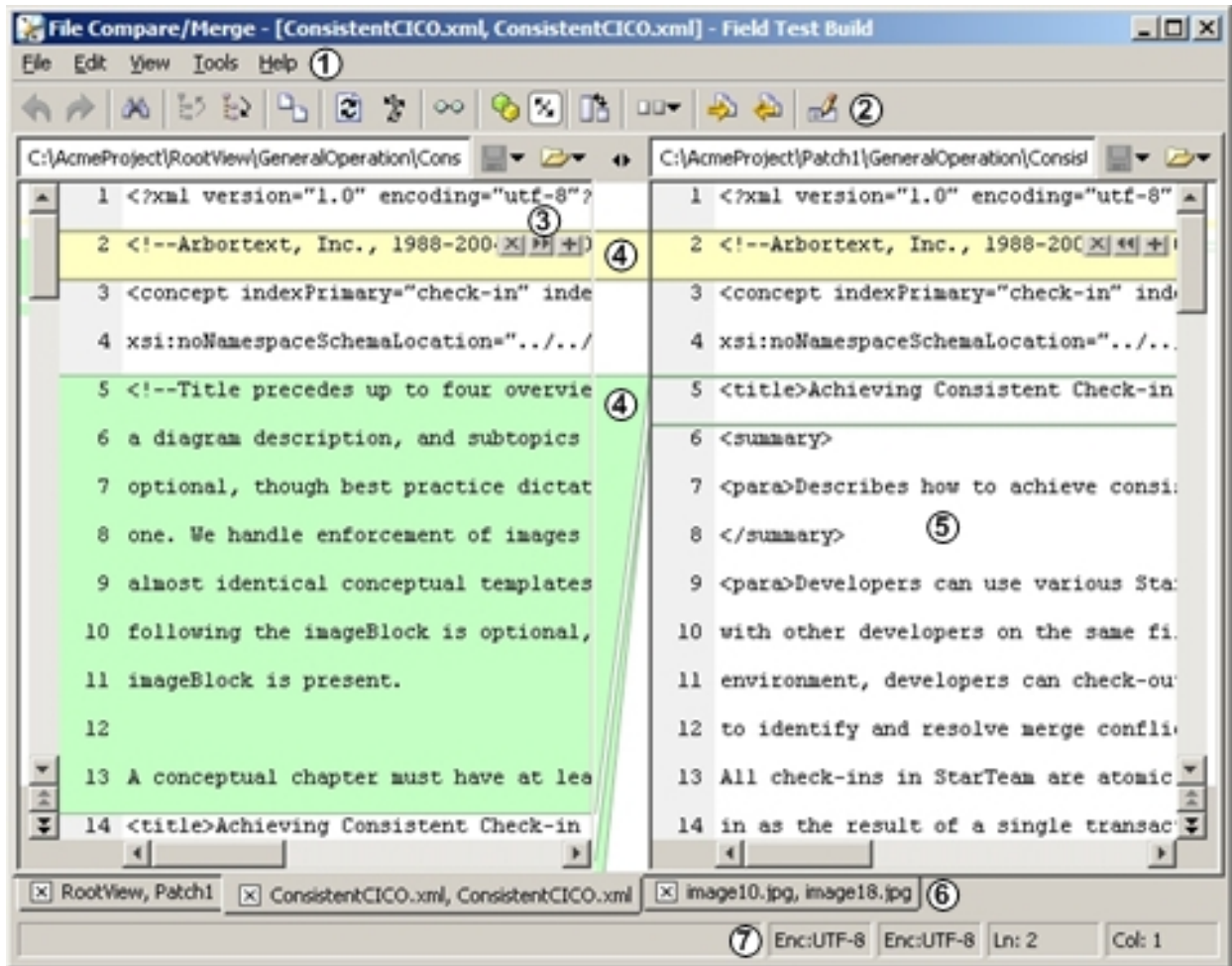
The embedded File Compare/Merge enables you to quickly compare the contents of a local text file and a repository file, or two repository files. This view does not allow editing or merging.



① Upper Pane ② Folder Tree ③ Lower Pane ④ Embedded File Compare/Merge Panes

Main File Compare/Merge

The main File Compare/Merge in the client enables you to compare local files with repository files, and edit or merge the contents. Each compare/merge session is displayed on a separate tab in the File Compare/Merge window.



- ① FCM Menu
- ② FCM Toolbar
- ③ Dynamic Edit-action Buttons
- ④ Difference Color Blocks
- ⑤ FCM Pane
- ⑥ FCM Session Tabs
- ⑦ FCM Status Bar

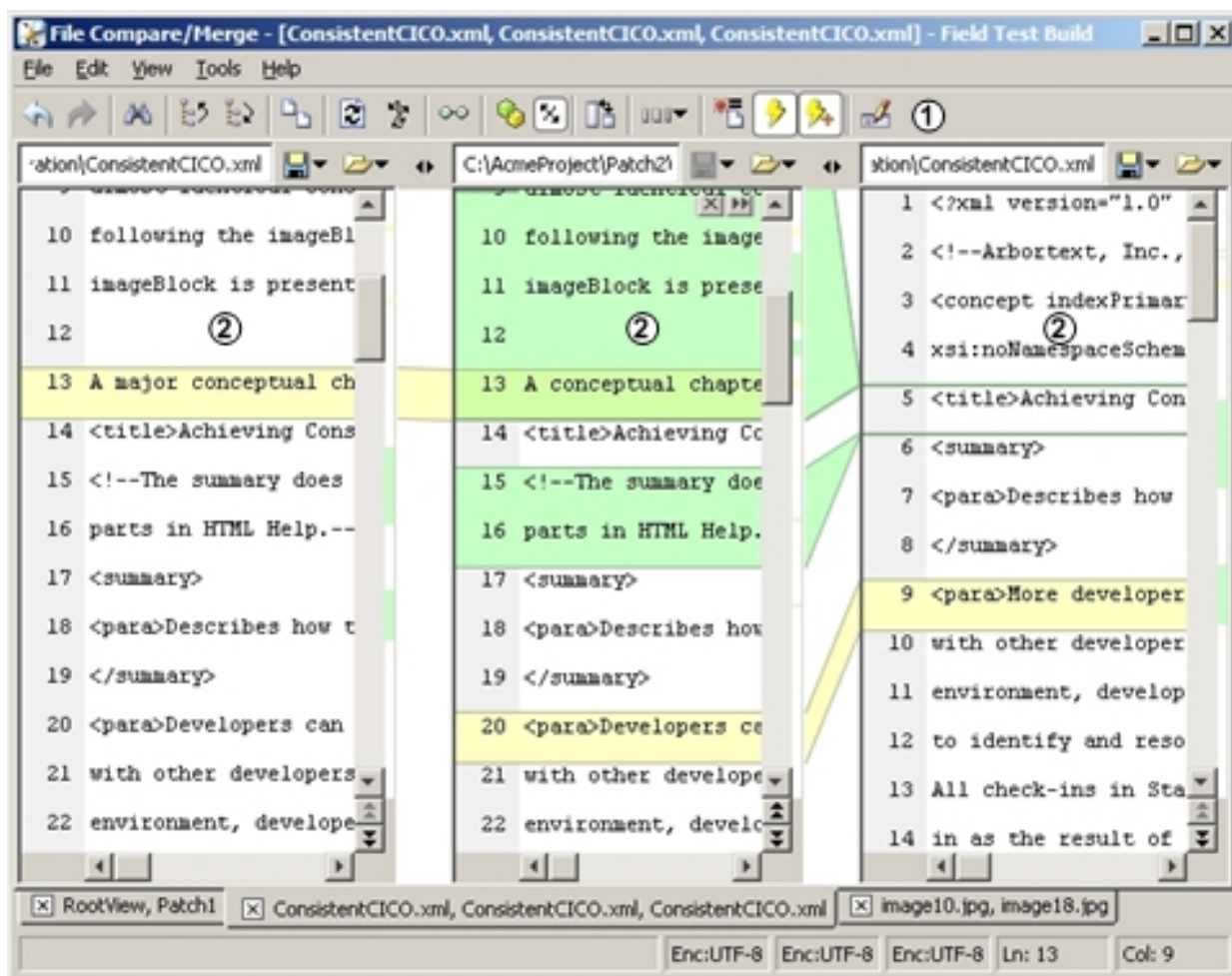
Standalone File Compare/Merge

The standalone File Compare/Merge runs outside the StarTeam client and provides the following features:

- ◆ A two or three-way file compare/merge.
- ◆ A two or three-way folder compare/merge.
- ◆ A two or three-way image comparison.

Three-way File Compare/Merge

The following image is an example of a three-way File Compare/Merge session.

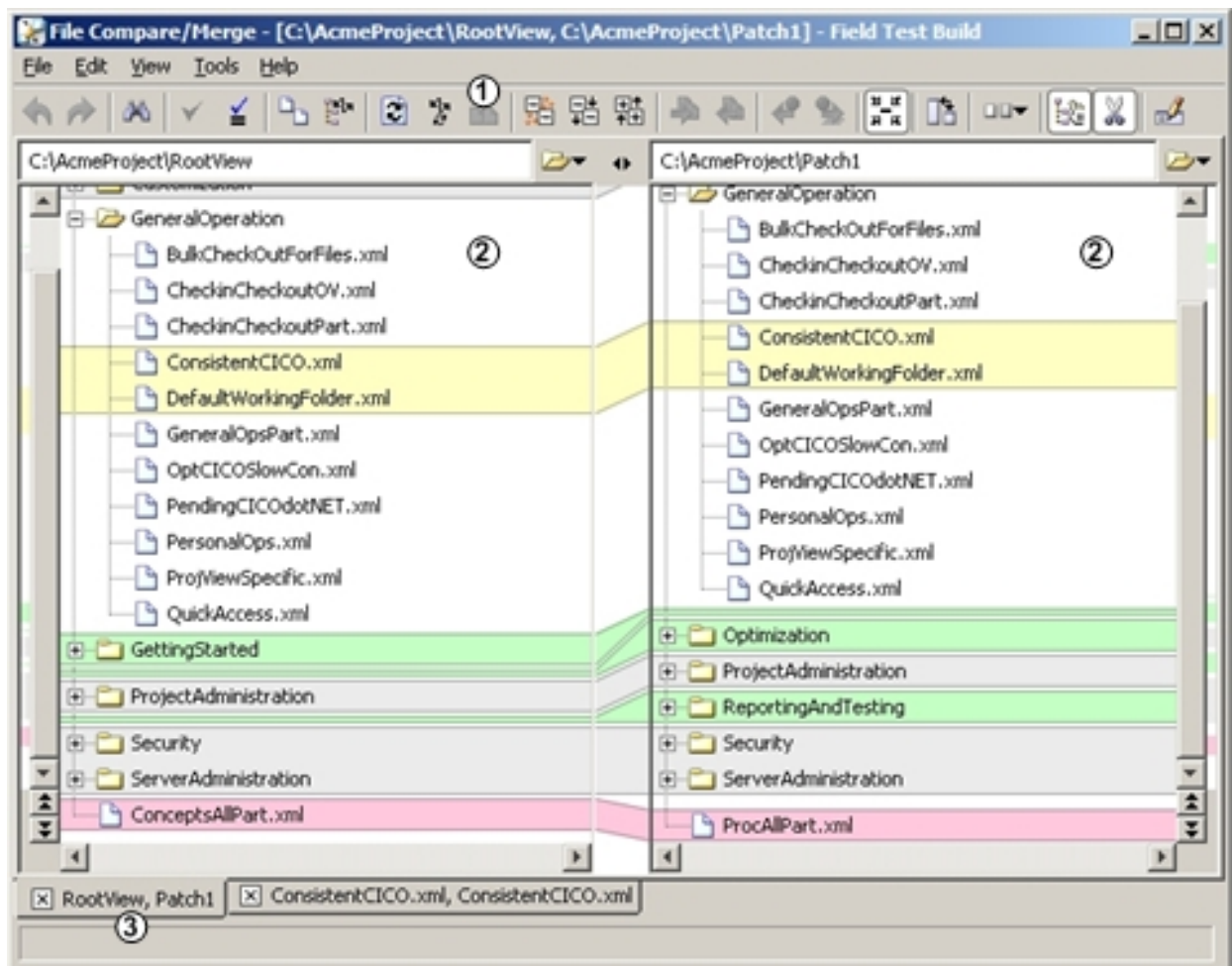


① FCM Three-way Merge Toolbar

② Three-way Compare/Merge Panes

Folder Compare/Merge

The following is an image of a folder compare/merge session.



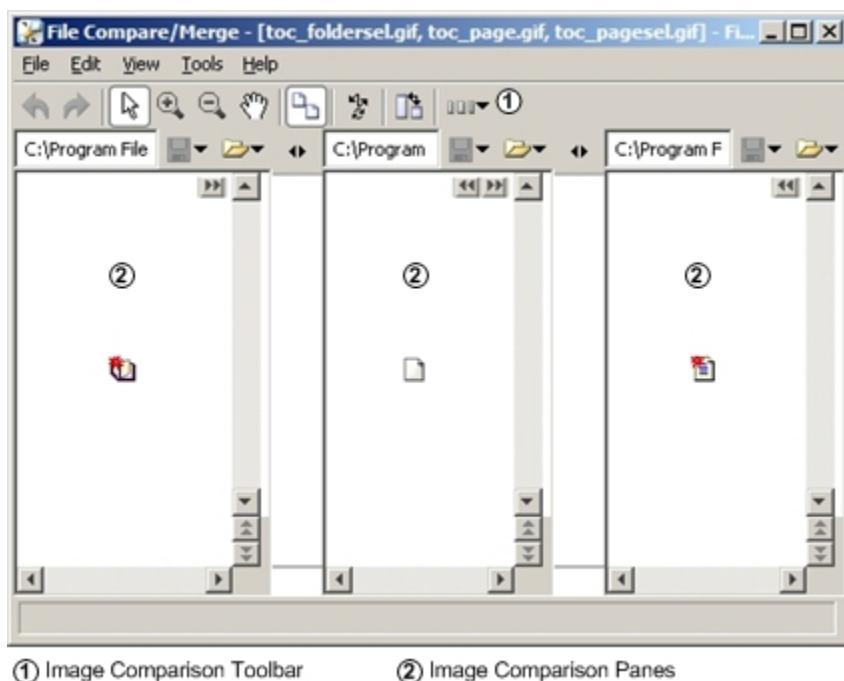
① Folder Comparison Toolbar

③ Folder Comparison Session Tab

② Folder Comparison Pane

Image Comparison

FCM has the ability to compare two or three images, and move or swap them, but not to edit them.



FCM Menus and Toolbar

The main and standalone File Compare/Merge windows have a main menu, a context menu, and a toolbar. The embedded File Compare/Merge has a context menu only, no main menu or toolbar.

The menus provide specific actions for comparing, editing, and merging files and folders. They are dynamic depending on which version of File Compare/Merge you are in, what type of files you are comparing, and how many panes you are using. In the main and standalone File Compare/Merge windows, a toolbar provides buttons for frequently used commands.

File Comparison Panes

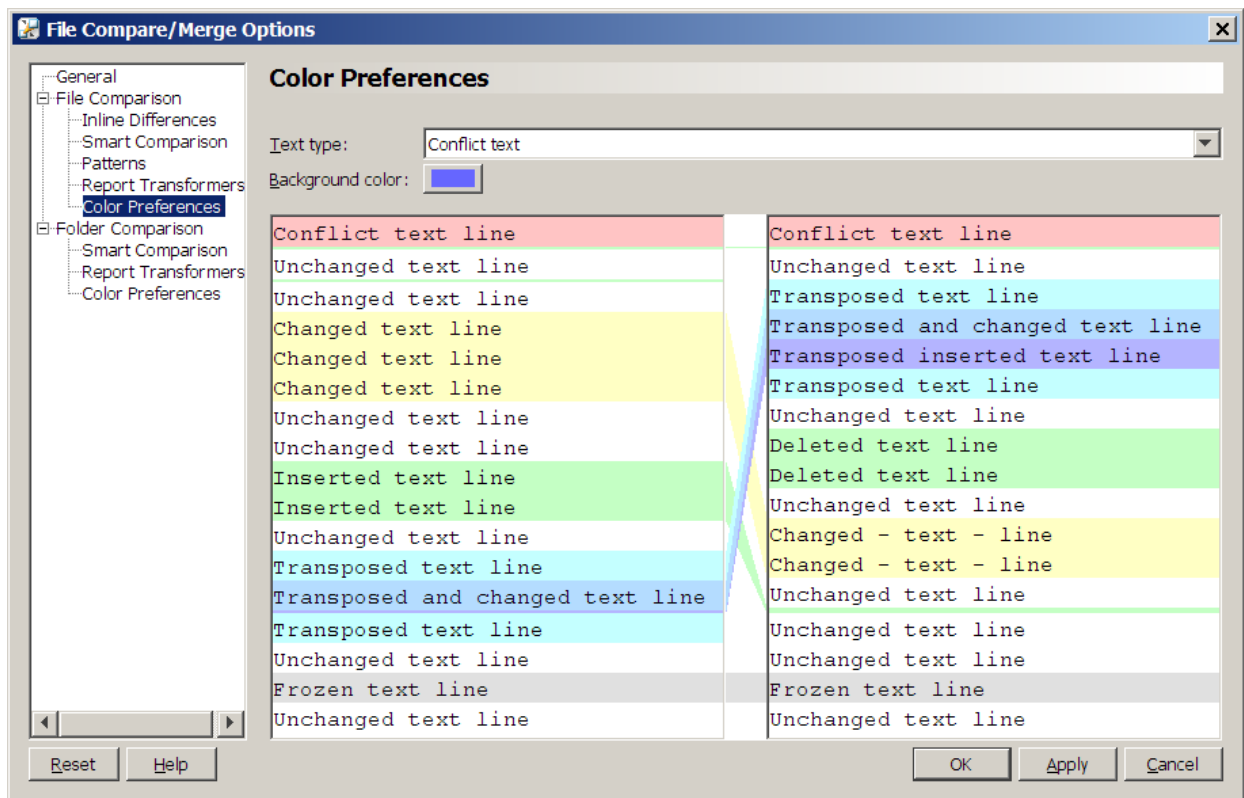
The file comparison panes contain the contents of the files or folders being compared. In the main and standalone File Compare/Merge windows, dynamic action buttons for editing appear on color differences blocks. You can choose to display a two-way or three-way vertical or horizontal comparison mode.

Colors Representing Difference Types

File comparison and folder comparison in View Compare/Merge highlight difference types in color, with each color representing a certain difference type. For example, it highlights inserted or deleted items in green, changed items in yellow, and moved or renamed items in blue. You can modify the default color for differences in the File Compare/Merge Options dialog box, accessed from the File Compare/Merge window under **Tools** ► **Options**.

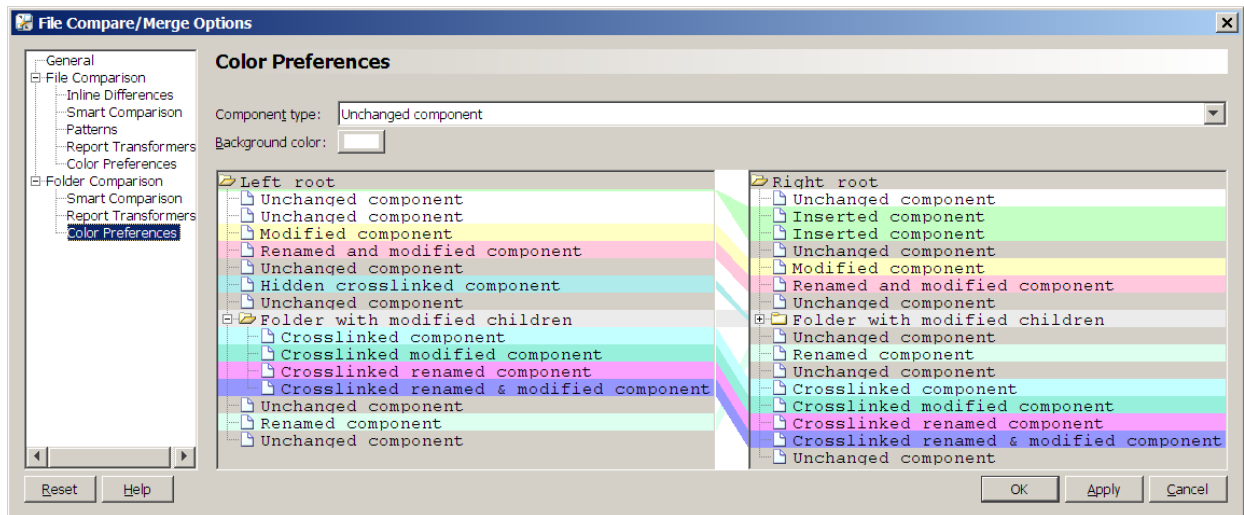
File Comparison Difference Colors

The following image shows the default colors used for highlighting file comparison differences in the **Compare Pane** of View Compare/Merge, and in **File Compare Merge**.



Folder Comparison Difference Colors

The following image shows the default colors used for highlighting folder comparison differences in the **Compare Pane** of View Compare/Merge, and in **File Compare Merge**.



Related Procedures

- [Comparing and Merging Files](#)
- [Comparing a Local File with a Repository File](#)
- [Comparing Historical File Contents](#)
- [Comparing Two Local Files](#)
- [Editing Files in a File Compare/Merge Session](#)
- [Saving Files Modified in a File Compare/Merge Session](#)
- [Comparing Folders](#)
- [Comparing Images](#)
- [Merging a Local File with the Tip Revision](#)
- [Merging Two Local Files](#)
- [Merging Folders](#)
- [Generating Reports from a File Compare/Merge Session](#)
- [Customizing Compare and Merge Reports](#)
- [Setting File Compare/Merge Options](#)

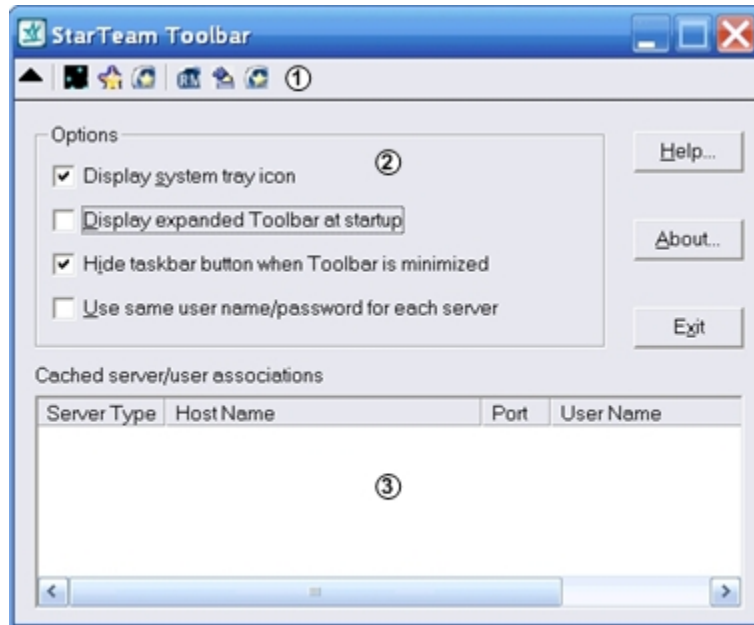
Related Reference

- [File Compare/Merge Options](#)
- [File Compare/Merge Keyboard Shortcuts](#)
- [File Compare/Merge Actions](#)

Toolbar Utility

This topic describes the UI for the Toolbar Utility.

Toolbar Utility UI



① Toolbar ② Options Pane ③ Cached Server/User Associations Pane

The Toolbar Utility is a component of the StarTeam and CaliberRM products which is designed to make it easier for you to log on to multiple servers and to launch different programs. The Toolbar Utility automatically caches the user name and password used to log on to each StarTeam or CaliberRM server, reducing the number of times you must log on to the same server to perform certain operations. The Toolbar Utility runs only on the Windows family of operating systems, and it is installed by default as part of the StarTeam Cross-Platform Client, and with the StarTeam Server to be used with the Server Administration Tool.

The rest of this topic describes the numbered components in the above image.

Toolbar

The toolbar is initially populated with shortcuts for the tools of the StarTeam and CaliberRM products installed on your workstation. Because the Toolbar Utility uses the standard Windows program shortcut feature, you can add any other program as a tool.

You can customize the toolbar by either:

- ◆ Using the context menu commands on the toolbar; or
- ◆ Working directly with the program shortcuts in the Toolbar Utility Tools folder, using an Explorer window.

For more information about modifying the toolbar, see the link “Modifying Tool Properties in the Toolbar Utility” at the bottom of this topic.

Options Pane

The options pane enables you to:

- ◆ Specify which portions of its user interface are visible (none, system tray only, window only, or system tray and window). By default, only the Toolbar Utility system tray icon is displayed.
- ◆ Use the same user name and password for each server, so you do not have to log on to other servers where that user name and password is valid.
- ◆ Add program shortcuts as tools on the Toolbar Utility, so you can easily start those tools. (You can also modify or delete an existing tool.)
- ◆ Display the Toolbar Utility help information or About dialog.

Cached Server/User Associations Pane

When you log onto a server and you select the **Save as default credentials for this server** in the **Log On** dialog box, the Toolbar Utility caches the user name and password you have used to log onto that server. (You can override the automatic log on feature when necessary. For details, see the link “Logging On to Servers with the Toolbar Utility” at the bottom of this topic.)

When you clear the **Use same user name/ password for each server**, you can save a different user name and password for each server. When you have the Toolbar Utility expanded, the lower portion of the window displays the cached server/user associations information about the currently cached server/user associations. The table displays the following:

- ◆ The **Server Type** column shows the type of server (StarTeam or CaliberRM).
- ◆ The **Host Name** column shows the network name of the machine where the server is located.
- ◆ The **Port** column shows the port number of a particular server configuration, so you can distinguish between multiple servers running on the same machine.
- ◆ The **User Name** column shows the user name portion of the default credentials saved for that server.

If necessary, you can resize the columns in the table or the entire window, to better display the current information. When you select the **Use same user name/ password for each server** option, the Toolbar Utility automatically uses the first saved user name and password for each additional server you try to access, so you do not have to log onto other servers where that user name and password is valid.

Note: Exiting the Toolbar Utility clears the cached information.

Related Concepts

[Where to Find Documentation for Each Product](#)
[StarTeam Product Overview](#)

Related Procedures

[Using the Toolbar Utility](#)
[Logging On to Servers with the Toolbar Utility](#)
[Modifying Tool Properties in the Toolbar Utility](#)

Concepts

This section contains all the conceptual topics.

In This Section

[Project Configuration](#)

Introduces a series of topics on configuring projects.

[View Configuration and Management](#)

Discusses the basic view types and their general characteristics.

[Overview of Folders and Paths](#)

General overview of folders and paths.

[Promotion States](#)

Describes promotion states.

[Personal Options](#)

Describes personal options you can set to customize the client application.

[Understanding Default and Alternate Working Folders](#)

Describes the Default and Alternate working folders and the results from changing them.

[Check-in and Check-out Operations](#)

This section contains conceptual topics related to the check-in and check-out process.

[Labels](#)

This section discusses the use of labels.

[Branching Options](#)

This section contains conceptual topics related to branching options.

[References to Folders and Items](#)

This section contains conceptual topics related to how StarTeam manages references to folders and items.

[Process Items, Process Links, and Process Tasks](#)

Discusses the enhanced linking model and the use of enhanced process items and process links, and process tasks.

[Comparing and Merging Files, Folders, and Views](#)

This section contains conceptual topics related to comparing and merging files, folders, and views.

[Atomic Check-ins](#)

This topic describes how atomic check-ins behave in StarTeam.

[Quick Access to Projects and Items](#)

Provides an overview of how StarTeam enables users to quickly access projects, views, folders, and items.

Project Configuration

This section discusses various issues around projects and their configuration.

In This Section

[Overview of Projects](#)

Describes projects and what they include.

[Keeping Projects Autonomous](#)

Present guidelines for keeping project autonomous by keeping cross-project links and shares to a minimum.

[Handling Cross-Project File Dependencies](#)

Provides techniques for more effective handling of cross-project file dependencies.

[Supporting Cross-Project Activities](#)

Describes how to handle lifecycle activities that span multiple projects.

Overview of Projects

A project is a way to group related items (such as files and change requests) hierarchically. Views and folders enable you to organize these related items more efficiently. For example, if you create a project for a software product, the files containing the product's functional specification, marketing requirements document, source code, and test suites can each be stored in separate folders.

Views can be used in a variety of ways. For example, different views can be used so that developers see only the project's source code folder and its child folders, marketing personnel see only the project's marketing folder and its child folders, and so on. In this case, each view has a different folder as its root. Views also support branching and parallel development

At the view level or item by item, you can branch data such as files and change requests. The branching enables you to create a special variation of your product. For example, you can start on the 2.0 version of your product without hampering the creation of service packs for the 1.0 version

You can create a project on any StarTeam server configuration, if your computer has access to that server and you have been granted the rights needed to create projects in that location. When you create a project, you must provide a project name and designate a working folder location for the project's root (initial) folder. The initial view of the project is created at the same time you create the project. It has the same name as the project, although you can change the name later if you choose. The root folder is also created at this time. If your computer is not currently set up to access the server on which the project will reside, you can add access to that server as part of creating the project.

Note: Beginning with StarTeam Server 2009, the server creates new projects with only the "File" type pre-selected as a default for new views. Users can still change the project properties after the project is created, and they can change the item types included for any given new view. However, if the user changes nothing, by default new views will only include files when they are created.

This change does not affect any existing projects. It only affects new projects created with new StarTeam Server 2009 Servers or existing servers once they are upgraded to StarTeam Server 2009. Adding other item types to the Project Properties (after the view is created) will NOT populate the items that were contained in the parent view (but left out during New View creation). If the user wants to bring the previous items into the new view, they must retrieve them by Rebasing from the parent view.

Related Concepts

[Project Configuration](#)

[Keeping Projects Autonomous](#)

[Handling Cross-Project File Dependencies](#)

[Supporting Cross-Project Activities](#)

Related Procedures

[Creating Projects](#)

[Managing Projects](#)

Keeping Projects Autonomous

The time-honored programming tenets of high cohesion and low coupling apply to StarTeam projects as well. The more independent your StarTeam projects are, the easier they are to secure, administer, and even separate from the original StarTeam configuration if necessary. Keeping projects autonomous means keeping cross-project links and shares to a minimum, avoiding them completely if possible.

Below are some guidelines for deciding what should be in the same project:

- ◆ A project should be used to manage the lifecycle artifacts of a cohesive application set or application component set. Examples are a commercial software product or a foundation library package. Multiple application or component sets can be managed in a single project if they are interrelated and generally enhanced and released together.
- ◆ A project should include all of the artifacts required to manage the lifecycle of the supported applications or components. This includes early lifecycle artifacts such as requirements documents, modeling diagrams, and design documents, as well as construction phase artifacts such as source files, images, and resource files; and later-phase artifacts such as test scripts and applications.
- ◆ • A project should include all artifacts authored in lifecycle phases as well as non-authored artifacts required to perform authoring. This includes, for example, all files authored by the IDEs such as workspace/project files, source code, and resource files. It also includes “input files” such as `.h`, `.lib`, `.jar`, or `.dll` files that are authored or generated elsewhere but required by the project’s IDEs or build processes. Input files may originate from third parties or from other projects in the same or other StarTeam configurations. (Transferring artifacts from one project to another is discussed further later.)
- ◆ Files directly generated from authored files such as `.obj`, `.class`, and `.lib` files generally do not need to be checked into the project. However, it is common practice to check in “final” binaries such as `.jar`, `.war`, and `.exe` files that are delivered to other projects, engineering test, QA, or other deployment phases. The need to place generated files under version control is highly dependent on your own development, testing, and release methodologies.
- ◆ A project should have a long-term view of the products or components it supports. That is, it should house all artifacts generated over multiple iterations through the lifecycle. This means that the project supports multiple versions of its applications or components, representing the complete history of those modules.
- ◆ StarTeam works best when “work” artifacts (CRs, tasks, topics, and requirements) related to a project’s files are stored in the same project. This allows, for example, a CR to be entered, tracked, and linked to the files in the same project to which the CR is related. This approach requires some special considerations for activities such as “CR triaging” and cross-project reporting. These issues are discussed later.

Some customers have attempted to use projects to separate development phases (for example, design and development) or development artifact types (like files and CRs). The artifacts are then interrelated by sometimes heavy use of links or shares. However, experience has found that copious use of shares – especially cross-project shares – results in difficulties in version control, reporting, security, and even performance. We suggest that artifacts related to the same applications and components—even though of different types and/or lifecycle relevance—should be managed in the same project.

Example 1: A Simple Client/Server Application

Scenario: A commercial software application consists of a server written in C++ and a single client, also written in C++. Furthermore, the client and server modules share a fair amount of source code and IDE projects that generate common DLLs. The client and server modules are generally enhanced and released together.

In this scenario, a single StarTeam project should be used to manage the combined files of both the client and server modules. The sharing of source code and shared release schedules suggest that the modules are cohesive parts of a single application. Requirements, design documents, CRs, and other artifacts required for all lifecycle phases of the client and server modules should be managed in the same project.

Example 2: An Independent Client Module

Scenario: A new Java client application is developed that uses the same server described in Example 1. Building and compiling the Java client requires some of the header files and one of the DLLs used by the server to produce a JNI wrapper, but no other source files. Furthermore, the Java application accesses other third-party servers and is generally enhanced and released on independent schedules from those used by the client/server modules.

In this scenario, it is reasonable to use a separate StarTeam project to manage the Java client's artifacts. The latest header files and generated DLL needed by the Java client are checked into a "external components" folder by the build process used in the client/server project. All CRs, tasks, and other lifecycle objects related to the Java client are managed in the same project.

Example 3: A Complex Financial Application Suite

Scenario: A complex application suite consists of a set of foundation components and nearly 100 separate applications, divided into five functional areas: accounting, insurance, forecasting, etc. The applications are developed by different teams and all use the foundation components, which are jointly maintained by the development teams. Applications within a functional area are highly interrelated, but applications between functional areas are fairly independent. The foundation component library is enhanced and released on its own schedule, but the entire application suite is released as a commercial product in coordinated releases.

Although the entire application suite is interrelated, multiple projects should be used due to the overall application suite size. The foundation components are managed in one project, and each of the five functional areas utilize a separate project to manage the corresponding applications (six projects total). The foundation project is enhanced, built, and then "deployed" to each of the functional area projects by checking in generated jar files. Each development team generally opens only one project to perform their normal work. However, a special build script (using the StarTeam SDK) is used to extract files from multiple projects and generate "whole suite" builds. The build script also automates the management of common view labels and promotion states across projects.

Related Concepts

[Overview of Projects](#)

[Handling Cross-Project File Dependencies](#)

[Supporting Cross-Project Activities](#)

Related Procedures

[Managing Projects](#)

Handling Cross-Project File Dependencies

When projects are highly cohesive, cross-project dependencies will be minimal, yet cross-project file relationships may still occur. Some files, either authored or generated, may need to be used in multiple projects.

The impulsive way to handle this situation may be to share the codependent files from one project to another. On the surface, this approach works, but experience has shown that cross-project sharing is problematic for several reasons:

- ◆ If a shared item's behavior is configured to "floating", changes from the parent project flow into the child project immediately, sometimes at inconvenient times. Many StarTeam users find that they need to manage the update propagation on a more scheduled basis.
- ◆ If a shared item's behavior is configured to a specific timestamp, it must be occasionally adjusted to allow updates to propagate. This makes the shared item read-only, and continually adjusting the configuration timestamp for a large number of shares can become burdensome.
- ◆ If a shared item's branch-on-change property is set to "true" (perhaps accidentally), and a change is made to the child share, the object will branch in the child project. This severs *automatic promotion* of changes from the parent item to the child share. If the child share is a folder, an innocuous change such as modifying the working folder will cause the folder object to branch.
- ◆ When an update is made to an item, the entire share tree must be locked. As the share tree grows, update performance is impacted proportionately.
- ◆ Normally, when obsolete views and projects are deleted, the StarTeam purge utility can be used to return unused database records and archive files. However, shares used by a deleted project or view can prevent the purge utility from achieving the expected reduction in database and archive size. In short, it might not be possible to reduce the size of servers that use cross-product sharing.

Because of these reasons, other techniques have proven to be more effective at handling cross-project file dependencies. Below are some alternatives to sharing that work in different situations. In the end, shares still may be the most appropriate way to expose files from one project to another, but the approaches below should be considered first.

Deployment Approach

If a project "owns" a set of files that must be checked into other projects, you can establish a process in which the files are periodically "deployed". This means that the file set is checked into the target project(s) on an as-needed basis, perhaps by a script. Often, a build script is a good place to embed the deployment task, especially if the file(s) to be deployed are generated by the build. Keep in mind that checking in the same file multiple times does not (generally) increase the size of the vault—each unique file revision is only stored once.

Configuration Approach

Sometimes the codependent files don't need to be checked into each project, but they need to participate in a common process such as a build or delivery process. In these cases, a simple configuration file (for example, XML) can be constructed that defines the files that participate in the process. If the file is checked into an established location and updated when the configuration changes, then build, release, or other scripts can check out the configuration file, parse it, and base their processing on its instructions.

Link Approach

In lieu of shares, links can be used to connect objects across servers. Links do not possess many of the problems exhibited by shares, and they can be pinned, unpinned, and moved to refer to different object revisions. The downside of using links is that they are untyped and possess no properties other than a link comment to identify their purpose.

Related Concepts

[Overview of Projects](#)

[Handling Cross-Project File Dependencies](#)

[Supporting Cross-Project Activities](#)

Related Procedures

[Managing Projects](#)

Supporting Cross-Project Activities

Regardless of how you partition your projects, you may find that certain lifecycle activities span multiple projects. Some examples of these scenarios and how they can be addressed are provided below:

Multi-project Builds	Build procedures that require files from multiple projects can use the StarTeam SDK, which can open multiple projects at the same time. Alternatively, iterative calls to the StarTeam command line tool (stcmd) can be used to check out files from each of the required projects.
Defect Triaging	When a new defect is discovered, it is often entered as a CR before the module that must be repaired is known. This means that, if CRs are normally managed in the project containing the files that will be modified, a paradox exists in determining where to create the CR in the first place. A project leader or other person usually “triages” the CR by assigning it to whoever he or she thinks should handle it. As the CR is analyzed and addressed, the module that is ultimately modified to address it may not be known for awhile. A simple solution for this scenario is to place all CRs in a well known place (perhaps a “new defects” project) and “move” (not copy) them to the appropriate project as needed.
Cross-project Reporting	Currently, StarTeam does not provide built-in, cross-project reports. Consequently, if you want to generate reports such as “all open CRs across all projects” or “cross-server file metrics”, your best option is to use StarTeam Datamart to harvest and report on CRs from multiple projects and even multiple configurations. Another option is to use the StarTeam SDK to write custom reporting applications.

Related Concepts

[Overview of Projects](#)

[Keeping Projects Autonomous](#)

[Handling Cross-Project File Dependencies](#)

Related Procedures

[Managing Projects](#)

View Configuration and Management

A view is a window through which you can access a subset of the artifacts in a project. Furthermore, a view uses items to reference artifacts and manage access to them. Depending on how it is configured, a view can serve as a subproject for a specific development or maintenance activity; it can be a read-only or updateable subset of another view; or it can be used for other purposes. The good news is that views have tremendous flexibility to serve many needs. The bad news is that you can get unexpected or undesirable results if you don't understand how the different view types work.

This section describes each view type, how it behaves, and when you might want to use it. It also discusses the roles views can fulfill for specific development activities and suggests practices for managing changes within a view and for propagating changes from one view to another.

In This Section

[Overview of Views](#)

Explains Views and how to use them.

[Understanding View Types](#)

Describes the various view types.

[View Roles](#)

Describes the roles of the various view types

[Proper Use of Views](#)

Presents general guidelines for creating and managing views.

[Change Management within a View](#)

This section describes best practices for managing changes *within* a view.

Overview of Views

When you create a new project, the server creates an initial or root view of that project with only with only the "File" type pre-selected as a default for new views. Users can still change the project properties after the project is created, and they can change the item types included for any given new view. However, if the user changes nothing, by default new views will only include files when they are created.

This initial view, which has the same name as the project, consists of the root folder, to which you will add child folders, files, and eventually, more. It is always read/write. The root view is ideal for collaborative development, because it is dynamic, showing all items in the project as they change.

To accommodate both user and project needs, however, the application enables you to create additional views of a project based on this view. These additional views, called child views, may contain some or all of the contents of the original view and may behave differently. For example, you might use child views to:

- ◆ Implement the same folder hierarchy for multiple releases of a product. If the root view is for current development, you maintain the folder hierarchy there and create new child views for each release at about the time that it ships. Maintenance would be done in the child view.
- ◆ Limit the portion of the project that certain team members see. Developers might need to see only the project's source code folder and its child folders; marketing personnel might need to see only the marketing folder and its child folders; and so on. Each of these views can have a different folder as its root.
- ◆ Support branching and parallel development. By branching files and other data in a new view, your organization can start to work on the 2.0 version of a product without hampering the creation of service packs for the 1.0 version.

Views represent configurations of items and support different development baselines of the same code base. It is common practice to promote changes from one release's maintenance view to the root view where the next release is being created. Views can be reconfigured to show items as they existed at an earlier point in time, or based on a view label or associated promotion state. Rollback views are read-only, as they show a precise state of the items and no longer permit changes.

Understanding Branching Views

A branching view is a view that permits branching—that is, the folders and other items in the view can branch or separate from the corresponding items in the parent.

Branching views serve many purposes; you can create a branching view to meet different needs from those of your main line of development. For example, you might create a branching view for a maintenance release or a custom version of your product. A branching view can also be used to keep an area of your project private until it is completed and tested. Then you can merge your changes into the main line of development when and where necessary.

A branching view should use a different working folder than that of its parent view. Using the same working folder for both views is not only confusing but can create status problems.

Item Branching Behavior

Given the appropriate settings, folders, files, and change requests in a child view can be branched—that is, can be separated from the corresponding item in the parent view. Folders and change requests branch when their properties change, while files branch when either their contents or their properties change. (Requirements, tasks, and topics can never branch.)

For each item, branching occurs a maximum of one time per view. For example, if a new item is added to the root view, its first revision has the dot notation 1.0. Subsequent revisions become 1.1, 1.2, and 1.3. Suppose this item is included in two child views created from the root view and that both of the child views are branching views. In one child view, if a new revision is made to the item, the item branches. This separation from its "parent" item in the root

view is indicated by an addition of two numbers to the dot notation. If the parent item was 1.3, the child item becomes 1.3.1.0. The child item's next revision becomes 1.3.1.1.

Now, suppose the corresponding item in the second child view is changed. Its dot notation must also change. Because 1.3.1.0 already exists, the separation of this item from its parent item gets the dot notation 1.3.2.0. This child item's next revision becomes 1.3.2.1.

The original item in the root view has the history: 1.0 > 1.1 > 1.2 > 1.3. The next revision will be 1.4. The item in one child view has the history: 1.0 > 1.1 > 1.2 > 1.3 > 1.3.1.0 > 1.3.1.1. The next revision will be 1.3.1.2. The item in the other child view has the history: 1.0 > 1.1 > 1.2 > 1.3 > 1.3.2.0 > 1.3.2.1. The next revision will be 1.3.2.2.

Whether or not a folder, file, or change request has the ability to branch depends upon its behavior settings.

- ◆ If the **Branch on Change** check box for an item is enabled and selected, the item can branch.
- ◆ If the **Branch on Change** check box for an item is enabled but not selected, the item cannot branch, but its behavior can be changed.
- ◆ If the **Branch on Change** check box for an item is disabled, the item has either already branched or its location is where it was added to StarTeam.
- ◆ If the **Branch on Change** check box for a folder is enabled but not selected, new items cannot be added. If you attempt to do so, an error message will appear, stating that the folder is read-only.

Note: Changing a folder's branching behavior does not affect the behavior of items in the folder; items in a folder have their own branching behavior.

Items in Both the Parent and the Branching View

Depending on the folders included in the new branching view, certain items from the parent view appear in the new branching view. However, these "inherited" items will have no labels, as neither view labels nor revision labels move from view to view.

As a result, the workflow for change requests is affected in the following ways:

- ◆ If the Last Build Tested and the Addressed In Build fields in a change request have no values when the change request branches, their workflow is specific to the new view only.
- ◆ If the Last Build Tested and the Addressed In Build fields in a change request have build labels as their values (i.e., these fields are not empty or do not contain the value "Next Build") when the change request branches, the branched change request retains those values. In the new view, these values can be changed, but only to the names of build labels that exist in the new view.
- ◆ If the Addressed In Build field contains the value "Next Build" when the change request branches, the "Next Build" value is replaced by the name of the next build label created in the parent view, not the next build label created in the new view.

Related Concepts

[Understanding View Types](#)

[Proper Use of Views](#)

[Change Management within a View](#)

Related Procedures

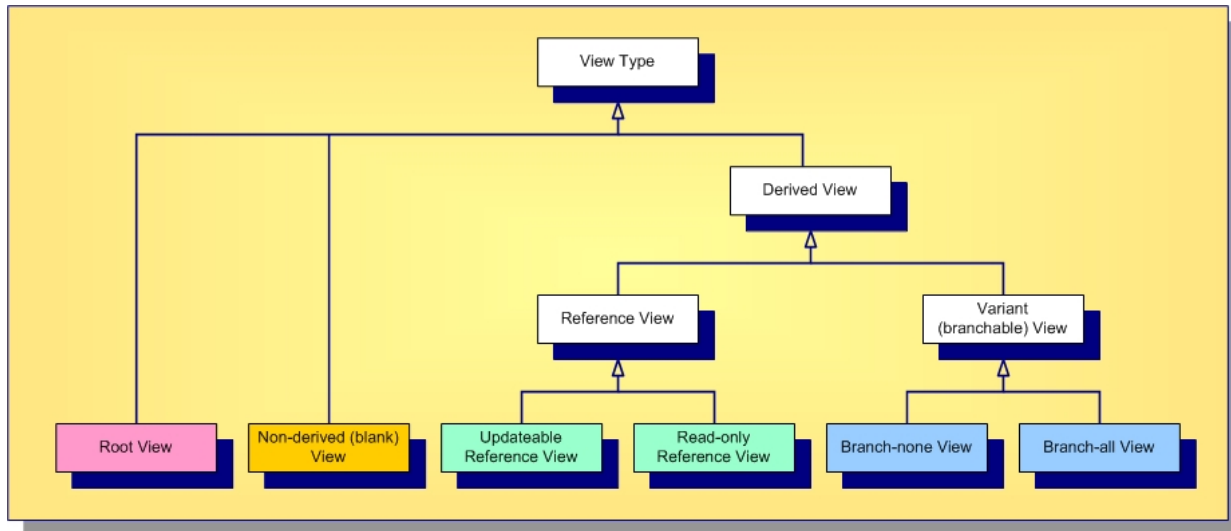
[Managing Views](#)

Related Reference

[View Type Options and Settings](#)

Understanding View Types

As shown, there are six “concrete” view types. Aside from the special *root* view type and one *non-derived* view type, the rest are *derived* view types, which subdivide into *reference* and *variant* types, each of which have two concrete types.



The abstract and concrete view types are described below.

- ◆ **Root View:** A root view is the “main” view automatically created for each project. Initially, it receives the same name as the project, but you can change it. (It is often renamed “Main” to emphasize its role.) When first created, the root view has only a root folder whose name matches the view. It is the only view type that has no *parent*—it forms the top of the project’s view hierarchy.
- ◆ **Non-derived View:** This is also called a “blank” view. Like the main view, it initially has no items except for a root folder. Although a non-derived view has a parent view, it is not *derived* from that parent, which means it does not inherit the parent view’s items. You can add new items to it, or you can share items from other views into it. A non-derived view can be used for non-lifecycle activities, acting like a “scratch pad” that you build up one item at a time.
- ◆ **Derived View:** A derived view begins life as a subset or an exact copy of its parent view. One of the parent view’s folders (often the root folder) is chosen as the root folder of the derived view; hence the derived view starts as a “window” into the artifacts at which it is rooted. What you can do with the artifacts in a derived view depend on whether it is a variant or reference view.
- ◆ **Reference View:** A reference view is a derived view that is a pure subset of its parent view. A reference view does not have its own items; it uses the same items as its parent view. Consequently, updates made to a reference view (if allowed) are applied to the same items used by the parent view. This also means that reference views never have their own artifact branches; hence reference views are also called “non-branching views”. Reference views also do not have their own view labels—they share the same labels as their parents. Reference views are considered lightweight since they do not have their own items. A reference view can be updateable or read-only.
- ◆ **Updateable Reference View:** An updateable reference view is a pure subset of its parent view with no added restrictions. If an item is updateable in the parent view, it is updateable in the reference view as well. Since the two views share the same items, changes in either view are immediately visible in both views. An updateable reference view is useful for exposing a portion of another view for security purposes. Instead of adding folder- and/or item-level security rights to the parent view, adding view-level security rights to an updateable reference child view is often easier to manage.
- ◆ **Read-only Reference View:** A read-only reference view is a subset of items from the parent view that cannot be modified through the child view. A child view can *float* to the tip configuration of the parent view. In this case, it immediately reflects any changes made in the parent view to items it can “see”. Alternatively, a read-only

reference view can be *pinned* to a specific configuration of the parent view: a timestamp, view label, or promotion state. A pinned read-only reference view reflects the state of the parent view (subset) as of that configuration. Once the child view is created, you can't change the configuration on which it is based. However, in the case of labels and promotion states, the child view will "follow" changes made to those objects. For example, if new items or different revisions are attached to the label or promotion state in the parent view, the child view will immediately reflect that change. Floating and pinned read-only reference views are useful when you want to subset a view and ensure read-only access, for example with applications such as build scripts.

- ◆ **Variant View:** Unlike reference views, a variant view is not a pure subset of its parent view. Although a variant view may initially be created as an exact copy or subset of its parent view, it has its own items. In fact, when a variant view is first created, the parent view's items (or subset thereof) are shared (copied) to the child view, initially referencing the same artifacts. When new items are added to the child view, they may not be automatically added to the parent view depending on the containing folder's configuration. Furthermore, since it has its own items, the variant view's items may be independently configured, which means they could branch. Consequently, variant views are also called "branching views". Whether or not a variant view's items are initially marked branch-on-change (BOC) is the major difference between the two variant view subtypes.
- ◆ **Branch-none View:** If a variant view's items are initially configured with BOC set to "false", it is referred to as a branch-none view. It acts a bit like a reference view, except that it doesn't share labels with its parent. If the child view's items are created with a pinned configuration, they will be read-only. If the item configurations float, updates through those items will float to the parent. This is why a branch-none view is sometimes called a "floating view". However, because the child view has its own items, "moves" in the child will not propagate to the parent view. Because of the discrepancy between the propagation behavior of moves versus other updates, branch-none views can be very confusing. It is possible to create branch-none views and hand-tweak item configuration and branch-on-change, but this approach has proven tricky at best and sometimes disastrous. Consequently, we recommend against using branch-none views.
- ◆ **Branch-all View:** Branch-all variant views are the most commonly-used views. A branch-all view begins life as a copy of its parent view (or subset thereof), with branch-on-change set to "true" for all items (that point to branchable artifact types). A branch-all view's item can be configured to float, causing changes in the parent view to float to the child view. However, branch-all views are far more useful when all items are configured to a specific timestamp or view label.

Of the six StarTeam view types, you will use main views and branch-all views with configured items (not floating items) the most. As detailed in the next section, each project will have only one main view, but you will use branch-all views to support both new development and maintenance activities. Reference views are occasionally used as a way to expose a read-only or updateable subset of its parent while simplifying security management. Blank views are rarely used, and branch-none views are not recommended.

Related Concepts

[Overview of Views](#)

Related Procedures

[Managing Views](#)

[Creating and Configuring Views](#)

Related Reference

[View Type Options and Settings](#)

View Roles

Technically, you could manage a project's artifacts using only the root view. For example, you could: add all new folders, files, CRs, and other items directly in the root view; acquire locks as you modify items to prevent conflicts between users; employ revision labels and process items to identify revisions related to the same logical change; create view labels to mark milestones such as specific builds and releases; and Use promotion states to propagate snapshots represented by view labels through a coordinated test and release process.

When your project is first getting started, or if you have a very small (and Safe) change to make, direct modification to the main view is fine. Otherwise, direct modification increases the risk of “breaking the build” if you haven't staged your changes elsewhere first. The moment your team needs to work on two versions of the same file, module, or application at the same time, it needs a place other than the root view to do its work. This is the purpose of *development streams*. You need containers that support parallel development or maintenance activities that require different artifact branches within the project. The child views you create under the root view provide these containers.

Because version control and SCM products have existed for several decades, it should be no surprise that many different *patterns* have been developed for managing software development artifacts. These patterns affect the number of development streams you deploy and how you propagate changes between them. With StarTeam, this translates to the number and type of views you use and how they are organized.

Most of the child views in your view *hierarchy* should be branch-all views. To determine when you need a new view, where it should live in the hierarchy, and whether a different type of view would be more appropriate, you should consider the *role* that each view will fulfill. Based on our experience, the best way to use views is to consider the roles described in the next sections.

Main View: Home Base for Artifacts

The main view, also known as the root view, should contain the latest, approved revisions for the whole project. By latest, we mean that the main view should match your latest changes, ready for the next release. By approved, we mean that it should contain revisions that have undergone whatever verification checks your process requires: a complete build, unit testing, integration testing, etc. By whole project, we mean that it should not be a subset of the project's modules. In short, your main view should contain the latest, complete, production work, ready to be seen and used by everyone.

Implicit in this recommendation is that the main view should always be clean: buildable and able to pass most if not all tests. Experience has shown that you will avoid many headaches by keeping your main view clean. To do this, all except for the simplest changes should be made in other views where they can be tested and fine-tuned until they're ready to be propagated to the main view. Once a new or modified revision has passed the point of no return—you're sure you won't change your mind—only then should it appear in the main view.

There is a subtle but important interaction between the main views and “share trees” you should be aware of. In general, the main view should contain the main (1.n) branch of each artifact. This happens automatically when you add a new artifact in the main view. StarTeam first creates a new 1.0 *artifact* and then connects it to the view and parent folder with a new *item*. But suppose you create a new artifact in a child view and then share the item “up” to the main view. Since the child view item was created first, it will be the “root share”, and the main view item will receive a “child share” item (initially pointing to the same artifact). In other words, the item share tree will point in the opposite direction as the views. You probably won't notice a problem until you modify the item in the main view, causing it to branch, thereby pointing to a non-main branch (e.g., 1.n.1.m). In future attempts to propagate changes between the two views, you'll find that it gets harder to propagate changes correctly because the share tree is backwards. For example, “rebase” operations won't work correctly. What can you do about this? The solution is to use the view compare/merge (VCM) facility to promote new items to the main view. VCM understands the share tree issue and propagates new items using an operation called *reverse share*.

Note: Savvy StarTeam pre-2006 users are aware of this subtle “share tree direction” issue and employ custom solutions. Some customers simply add new items to the root view first and then share them “down” to child views. Other customers propagate new child view items to a parent view by first moving them up and then sharing them back down – this is a basic version of what a VCM reverse share does.

Activity View: Isolated Team Work Area

Suppose your team is charted with developing a significant new enhancement (or a set of enhancements to be developed together). By “significant”, we mean changes that will affect more than a few artifacts, take more than a few days, and/or involve more than a few people. You wouldn’t want to make changes directly to the main view since (a) you want to check in your changes often to ensure they are backed-up, progress can be tracked, etc., and (b) it may be a while before your changes are sufficiently stable to begin testing. The way to isolate your team’s work is with an *activity view*.

An activity view is a branch-all view created from a well-identified, stable configuration of the main view. If the enhancement work only requires a portion of the main view’s modules, you might choose to root the activity view from something other than the main view’s root folder. Typically, an activity view is created from a view label or promotion state, which causes it to contain items pinned to the same revisions in the main view as of that snapshot. As changes are made to the activity view, the corresponding items will branch and hence will not be visible to the main view. New items are also added to the activity view and hence not visible to the main view. As work progresses, the state of tip revisions in the activity view may not always represent a buildable release. But eventually, your team will build, test, fix, and finish its work, whereupon it is “promoted” to the main view.

Activity views typically have a limited lifespan: when the enhancement work is done and promoted, they can be deleted, usually after a certain period of time. Strictly speaking, a single activity view could be used for multiple enhancement activities, but there is an advantage to using separate views for each activity: if for some reason the activity must be cancelled, the activity view can be abandoned and eventually deleted. This isn’t practical if the view contains work from multiple activities. Either way, activity views periodically require “rebasing”.

A variation of the activity view is an *integration view*, which supports integration activities for a large software project.

Release View: For Post-Release Maintenance Work

When you release a snapshot of your software, you’ll probably have to maintain a development stream just for hot fixes, patches, service packs, and so forth. A view that fills this role is called a *release view* (or a *maintenance view*).

Like an activity view, a release view is a branch-all view created from the main view as of a specific snapshot. A release view is created after one or more enhancement activities have been completed and promoted to the main view. It represents a milestone where your software has been (or is about to be) released externally. The release view is almost always rooted at the main view’s root folder, and it is usually created from a frozen view label to clearly establish the software configuration that was actually released. In fact, many organizations create the release view first and **then** build and deliver the software from the release view.

Since it is a variant view, a release view can receive changes: for example, to fix bugs. If these bugs must be propagated to the main view, they are promoted as in activity views. However, release views are generally not rebased to receive changes from the main view except for bugs fixed in the main view first that must be applied to the release view as well.

If you need to make significant changes to a release view (perhaps a service pack), it is acceptable (even advisable) to create an activity view as a **child** of the release view. You would then make the changes in the activity view, perform appropriate validation tasks, and then promote it to the release view. You might even want to use a sandbox view, which is discussed below.

Sandbox View: For Small Teams or Individuals

A *sandbox view* is similar to an activity view in that it provides a place where changes can be made without disrupting the parent view. A sandbox view is also a branch-all view created as a child of the view to which its changes are intended to be promoted. However, a sandbox view is different than an activity view in the following ways:

- ◆ A sandbox view is intended for use by a single individual or perhaps a very small team. Though not mandatory, the view's permissions could be set so that only authorized users (and administrators) can open it.
- ◆ A sandbox view is often created for work on a subset of the overall project such as a specific module for which its owner is responsible. Consequently, its root folder may be created from a non-root folder in the parent view. This makes the view faster to open and easier to navigate in graphical clients.
- ◆ A sandbox allows (and encourages) frequent file check-ins so that work is saved and can be tracked. The formal or informal rules for making changes are often relaxed. For example, checked-out files might not be locked; builds might be performed only on demand; and full testing and validation might be performed only when changes are promoted to the parent view.
- ◆ As an alternative to the activity view, whose lifespan matches that of a fixed-length development activity, a sandbox view may be retained for a much longer period, spanning multiple tasks. In this role, a sandbox view is usually created as a child of the main view.

Sandbox views dedicated to individual developers are practical for small- to moderate-sized teams. For large teams, instead of creating thousands of personal sandbox views, it is more practical to use activity views that are shared by multiple team members.

Build View: Read-Only Windows for Build Scripts

Many StarTeam customers create build applications using simple build tools (for example: make, nmake), a commercial build product, or open source components (for example: Ant, CruiseControl). It is easy to integrate these tools with StarTeam due to the availability of its full-featured SDK and the availability of pre-built components such as StarTeam Ant tasks and a CruiseControl "bootstrapper" plugin for StarTeam. Build applications typically open a view, specify a snapshot configuration (timestamp, view label, or promotion state), and then check out the files they need.

In some cases, however, an organization may want to restrict the access of the build tool (or user) in comparison to the permissions other users have to the view. For example, you may want to guarantee that (a) the build tool can see only artifacts required by the build (and not design documents or other artifacts), and (b) the build tool has read-only access and cannot modify anything.

This situation could be handled through folder- or artifact-level security rights. However, this situation occurs often enough that some customers have found it useful to create a view tailored to the needs of the build tool. Such a *build view* is often created as a read-only reference view based on a promotion state. The reference view may be rooted at a non-root folder of the parent view. Consequently, a limited set of artifacts are exposed, which cannot be modified. Whenever the promotion state is assigned to a new view label, the new artifact revisions automatically "appear" in the build view. That is, the build view "follows" changes to the promotion state.

Because security is easier to administer at the view level, a build view is often a more efficient way to accommodate build applications.

Related Concepts

[Overview of Views](#)

Related Procedures

[Managing Views](#)

[Creating and Configuring Views](#)

Related Reference

[View Type Options and Settings](#)

Proper Use of Views

Irrespective of view type, below are some general do's and don'ts to consider when creating and managing views.

- ◆ Try to ensure that items in the main view refer to the main (1.n) branches of its artifacts. Also, be careful not to delete the 1.n branch of an artifact unless it has truly achieved end-of-life.
- ◆ “Leaf-most” views should be considered disposable. Eventually, when a view's corresponding development activity is finished or its corresponding maintenance release is no longer supported, it should be deleted. This is important to prevent projects from growing unbounded, and it keeps the project structure uncluttered and easy to navigate. Deleting leaf views provides the Server Administration tool “purge” process with the maximum opportunity to actually shrink the database and vault by removing unneeded data.
- ◆ Three or four view levels in a project are normal. If you find your project has more than that, you might not be following best practices for views. For example, you may be making the mistake of performing new development work directly in the main view, forcing other active views to keep spawning child views. Or, you may be neglecting to promote the latest and greatest changes to the main view.
- ◆ Don't use branch-none (“floating”) views except in extremely rare cases when you understand exactly how they work.
- ◆ You can “refactor” projects that have become too big by moving items in the main view to another project. Old CRs, tasks, and other projects can be moved to an “archive” project so they can still be accessed without cluttering the main view of an active project. If you have a project with numerous modules or applications, a large number of files can cause it to take too long to select “all descendants” in the main view. You can break the project up by creating new projects and moving (not sharing) folders and items corresponding to whole components or applications from the old project to the new projects. Do this when the main view reaches a major milestone such as after a new release. If you want to refactor the project by reorganizing the folder tree, do this before you split up the project. Create view labels before and after you refactor a project so you have markers for what changed.

Related Concepts

[Overview of Views](#)

[Understanding View Types](#)

Related Procedures

[Managing Views](#)

[Creating and Configuring Views](#)

Related Reference

[View Type Options and Settings](#)

Change Management within a View

Undoubtedly, your most active views will be activity and sandbox views. But, unless you've figured out how to write defect-free software, you'll also need to make changes to release views. And, although we don't recommend it as a general practice, for small changes, it often makes sense to perform updates directly in the main view. In all of these cases, your team has several choices for the process by which they perform updates.

The appropriate process depends on the size of your development team (which affects the probability of conflicts) and the level of detail with which you need to track changes. So, let's first review the ways in which changes are documented, tracked, and queried in StarTeam:

- ◆ Each new artifact revision can receive a *revision comment* where the user can document why the revision was created. Projects can be configured to require the revision comment for files.
- ◆ An artifact's revisions can be reviewed and compared in the History tab of the cross-platform client (CPC). Historic revisions include the revision comment, modifying user, modification timestamp, version number, and all other properties as of that revision. The StarTeam cross-platform client (CPC) allows you to customize what properties are shown in the History tab and to compare file content changes in a dedicated "file compare" pane.
- ◆ Unless audit generation has been disabled for the configuration, an audit record is generated for both revision-generating changes and other updates such as moving an artifact to a new folder or attaching an artifact to a label. You can query audit records in the CPC's Audit tab. The length of time that audit records are retained is configurable via the StarTeam Administration tool. We recommend that you always enable audit generation and retain records for at least 90 days.
- ◆ Through process items or explicit actions, links can be used to connect change items such as CRs and tasks to files that were added and modified on their behalf. Links generated via process items are pinned to the specific artifact revisions involved at both ends. You can view an artifact's links in the CPC's Link tab. To navigate to a linked item, right-click the item and choose **Select Linked Item** from the shortcut menu.
- ◆ Artifacts can be attached to revision and view labels so that revisions related to a common task or milestone are easily identified. (StarTeam can automatically attach new and modified files to a revision label at add/check-in time.) Choose **Select > By Label** to see the artifacts attached to a label. The labels to which each artifact revision is attached can be seen via the CPC's Label tab.
- ◆ The CPC offers several text-based reports and graphical charts for reporting on historic revisions, links, audit records, and other information. You can also create custom report templates that can be saved and reused.
- ◆ All of the change information described above can be accessed via the StarTeam SDK. If none of the built in reporting mechanisms are sufficient for your needs, you can build custom reporting tools or applications. Custom tools can even be centrally stored and automatically downloaded to CPC users, where they can be activated via the toolbar.
- ◆ The StarTeam Datamart product allows information from a StarTeam project, configuration, or multiple configurations to be extracted and loaded into a relational database, which can be queried and analyzed through reporting and business information tools.
- ◆ Borland Search is a search engine product that indexes and provides secure searching of multiple StarTeam (and CaliberRM) repositories. It can search across "space" (repositories, projects, views, and folders) and across "time" (historic revisions). All searching honors the security of the harvested artifacts; hence, it provides another way to find all the stuff your organization is storing.

Now that we've looked at the primary means by which you can report on changes within a view, let's look at ways in which you can manage change. We'll do this by examining three scenarios in increasing order of "formality".

Scenario 1: Working in a Small Team

If you're working on a project managed by a small team, you may not need much formal change management. Furthermore, if code modules are clearly assigned to individuals, or if you can just lean out your office door and yell,

“Hey, I’m working on the report module”, then the possibility for conflicts may be very low . In this case, your change process may be very simple:

- ◆ Maintain a set of local working folders where you keep the latest files for each view.
- ◆ Before you begin a new development task, get your working folders up-to-date by checking out any “Missing” or “Out of Date” files.
- ◆ As you modify files in your working folders, you’ll see their status in the CPC’s File tab become “Modified”. New files will appear as “Not in View” because they have not been added to the view yet. New folders will appear as “Not in View” in the Folder tab. If you enable **Show Not-In-View Folders** in the Folder Tree menu, you’ll see not-in-view folders in the folder tree regardless of which tab is selected. The folder tree always shows “Current” folders and those “Missing” from your workspace.
- ◆ When you’re ready to check-in, refresh your window (if you don’t have automatic refresh turned on) to see if another developer added or modified anything that may affect you. Check-out any new “Missing” or “Out of Date” files.
- ◆ If another developer checked in a file that you modified locally, its status will become “Merge”. The best way to resolve the merge conflict is to check out the file and answer “Yes” to the “Do you want to merge” prompt. This will launch the File Compare Merge tool. If there are no conflicts, the merge tool usually just saves the merged result file over your existing work file. Otherwise, you may be prompted to review and tweak the result file before continuing. Once saved, the file’s status will change to “Modified”.
- ◆ When you have only “Not in View” and “Modified” files, to do a build, run unit tests, and/or do any other validation steps your team requires.
- ◆ Finally, commit your changes to the view. You can select all of the new (“Not in View”) and modified files and check them all in at once. If you add and/or check-in multiple files in a single step, they are committed as an atomic transaction. Moreover, file content is pushed to the server “outside” of transaction state in a restartable manner. So, if you cancel a large multi-file add/check-in operation before it commits, you can restart it by just doing it again—only files that didn’t make it in the first attempt will be sent to the server. The commit happens at the end only when all content has arrived using a single transaction.

Note: One caveat of adding new files and checking-in modified files in the same check-in dialog is that the revision comment is also used to initialize the “description” property of new files. If you want accurate descriptions for new files, add each one separately or modify each file’s description after adding it. New folders are added automatically—you only need to explicitly add new folders on the server when they are empty on the client.

And that’s it. You get revision history and audit records automatically. This process doesn’t fool with labels, and you don’t get process item links either. But it might be all that your organization needs. As Einstein said, “Everything should be made as simple as possible, but not simpler.”

Scenario 2: Preventing Merge Conflicts

As your team size grows, the problem you may find with the previous scenario is encountering too many “unexpected” merge conflicts. Your team may want everyone to know when you’re about to make changes to a file before you actually do so. To better communicate intent and reduce the likelihood of merge conflicts, you can amend the process outlined in the previous scenario as follows:

- ◆ In the Personal Options dialog box, select the options “Mark unlocked working files as read-only” and “Clear file locks on check-in”.
- ◆ When you want to edit a file, lock it via the CPC first. (In some IDEs with StarTeam integrations, when you attempt to edit a read-only file, the IDE will offer to lock the file for you.) The lock will notify others that you’re editing the file.
- ◆ When you check-in modified files, the CPC will automatically unlock them for you (in the same transaction in which the changes are committed).

This process both informs team members of who is working on what, and it minimizes merge conflicts. Note that your team can require this process by setting the project-level options “Require exclusive lock when files are checked-in” and “Mark unlocked files read-only”. It’s up to you and your team to decide if you want this level of enforcement.

Scenario 3: Using Process Items

The next step in a more formal intra-view change management practice is to use process items so that all file modifications are linked to an appropriate change item. You can enforce the use of process items at the project level using the “Process Rules” options. These options let you:

- ◆ Require the selection of a process item when new files are added or modified files are checked in.
- ◆ Select which item types are eligible as process items. Your choices are CRs, tasks, and requirements.
- ◆ Specify whether or not the *status* of each item type will be considered in order to be used as a process item and which status values are permitted. For example, you can specify that CRs must have a status of “Open”, but tasks can be used regardless of status.

When you enforce process items, existing files cannot be modified and new files cannot be added until an eligible process item is selected. Consequently, new and modified files are automatically linked to the selected process item, enhancing the context information of these changes. Moreover, the links are created in the same atomic transaction in which the file updates are performed.

Note: In the add and check-in dialogs, you can elect to mark the selected process item “closed” (CRs), “finished” (requirements), or “complete” (tasks). If you choose this option, the process item update is also performed in the same atomic transaction.

Related Concepts

[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)

Related Reference

[View Type Options and Settings](#)

Overview of Folders and Paths

In StarTeam, three types of folders play important, yet dissimilar, roles.

Original workstation folder	Users set up this folder and its contents on a workstation, then use the New Project Wizard to create a new StarTeam project. This folder, which may contain files and other folders, becomes the root folder of the new project – that is, it becomes the root folder of the project's initial or root view. StarTeam creates the project, the root view of the project, and root folder at the same time. The project, view, and root folder initially have the same name, although the name can be changed later
StarTeam folder	These folders are used in StarTeam to group items in a project view. For example, a folder named Source Code can group source code files, requested changes to those files, and other related items. These folders can be created automatically at the same time as a project or added later by administrators or team members with the appropriate privileges. The hierarchy of StarTeam folders in the current view appears as a folder tree in the project view window.
Working folder on the workstation	<p>A working folder is actually a property of a StarTeam folder, but is quite different, as it is an object controlled by the operating system. It stores files that are copied or checked out from StarTeam or that will be added to StarTeam . A StarTeam folder is an object controlled from within StarTeam. Data about it is stored in the database that holds all project data.</p> <p>A project, its root view, and its root folder all have the same working folder. If additional views of the project are created, each view and its root folder have the same working folder. The working folder for the root folder always has an absolute path, which starts with the drive letter and lists the appropriate directories until it reaches the working folder itself.</p>

You can add to a StarTeam project at any time after it has been created. These folders can be moved from or shared with another StarTeam view or added from a workstation.

Understanding Working Folders

Understanding the relationship between application folders and their working folders is important because the working folder stores the files that you check in and check out.

Each folder has a default working folder from which you modify working files. For team members that use the same folders, the working folder structure on one person's workstation is often the same as those on another person's workstation.

When you check out a file, the application copies the requested file revision to the appropriate working folder. If the working folder does not already exist on your workstation, the application automatically creates it for you as you check out files that go in that folder.

The application expects you to add and check in new file revisions from those working folders. If the working folder does not exist on your workstation, you can create it manually or automatically using the Create Working Folders command. After the working folder exists, you can add files to it.

The exact location of a working folder is displayed as one of the application folder's properties.

Alternate Working Folders

The view's working folder may not be the optimal choice for all users. You, or any other user with the access rights to do so, can select a more useful location for the view's working folder on your own workstation by designating an alternate working folder. For example, you might want to use a shorter path or a different drive letter. Remember

that a working folder must point to a physically discrete location, such as a drive on your workstation or a personal directory on a shared file server. Borland does not recommend putting your settings on a mapped network drive.

The alternate working folder path for the view is specific to the workstation and user. For example, if you log onto the project as another user or use another workstation, your alternate working folder setting is not known.

When you designate an alternate working folder for the view, the path to the working folder for each child folder in the view may be similarly modified for your workstation.

For every folder in the hierarchy whose working folder is relative to the path of the view's working folder (as opposed to having an absolute path or an alternate working folder path of its own), your alternate path for the view's working folder becomes part of the paths to its child folders' working folders.






Folder Paths

StarTeam often stores working folder paths from development environment applications as relative paths. For example, `..\sc` may be the working folder for a project's [Source Code](#) folder. If you move a folder to another location in the StarTeam hierarchy, its working folder may end up in an unexpected location. This result occurs because the application applies the relative path to the working folder path for the new parent folder. Therefore, if you move a folder, you may want to specify a working folder path that is not relative, to avoid accidentally changing the working folder path on users' workstations.

Folder Tab

StarTeam includes a component tab for folders called **Folder**. When selected, this tab displays a main menu item and context -menu that contains many of the same menu commands that you would use when working with files, change requests, requirements, and so on. It is possible to perform some operations on multiple selected **Folder** items, such as adding files to a view.

Folder states are represented by the following folder icons:

	Regular StarTeam folder.
	Invisible Folder: Indicates a folder where the Visible property has been unchecked in the Folder Properties dialog box.
	Not-in-View Folder: Indicates a folder on your local disk that does not map to a StarTeam folder.
	Missing Working Folder: Indicates that local working folders do not exist.
	Folder uses an alternate working folder path than the default one set up by the project.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Folders to Views](#)

[Changing a View's Default and Alternate Working Folders](#)

Promotion States

A promotion state is a state through which a product passes. For example, most software products go through a release or production cycle – that is, the product moves from developers to testers and back again, until it is ready to go to the marketplace. Promotion states provide a convenient mechanism for ensuring that the right files or other items are available to the right people at the right stage of this cycle. For example, if a software administrator creates Test and Release promotion states, files that are ready for testing can be assigned to the Test state and files that have been tested successfully can be assigned to the Release state.

The promotion state feature permits an administrator to create promotion states and associate a view label with each state. An administrator creates a new promotion state configuration which is the basis for a new view or a reconfigured view containing only items with a specified promotion state. Administrators can also set access rights for promotion states. The view labels assigned to a promotion state are usually also used as build labels, so that they can serve as properties in change requests.

The view label for a state can be changed whenever appropriate. It can also be promoted from one state to the succeeding state. For example, although testers may always be using files in the Test promotion state, the files may be from Build 07 in one week and from Build 08 in the next. Users usually configure the project view for their job assignment by promotion state instead of by view label. For example, testers would configure their view to the Test promotion state.

Many features of the application depend on calculations involving times and dates. In particular, labels, configurations, and promotion states are all governed by time and date calculations. If the clients and the Server are not kept synchronized, a number of operations (such as checkouts, file status displays, or label creation) may fail or produce inaccurate or unreliable results.

Understanding Access Rights for Promotion States

Each view has its own set of promotion states. Access to these states is controlled by:

- ◆ The **Define Promotion Model** right which is set on the **View** node of the **Access Rights** dialog box for both projects and views. See “Granting Access Rights at the View Level”. A user with the **Define Promotion Model** right can do anything to the promotion model.
- ◆ Access rights that govern access to individual promotion states. These are **Generic Object Rights** and **Promotion State Specific Rights** which are set on the **Promotion State** node of the **Access Rights** dialog for both projects and views. They also appear on the access rights for individual promotion states.

The rights for an individual promotion state are checked at the state level; if necessary, the checking continues at the view level and eventually the project level. If a user is granted a given right at one level, there is no need to check the next.

- ◆ When a right is granted at the view level, it applies to all states in the view, unless access is denied at the state level.
- ◆ When a right is granted at the project level, it applies to all the states in all the views within the project, unless access is denied at the state or view levels.

Example of Using Promotion States

Suppose a software company wants to use the following promotion states to correspond with its use of the application:

Development	Developers work with the tip revisions of files. These files have no view label because they are constantly changing. Many companies do not use Development as a promotion state, because
-------------	---

configuring a view to a promotion state, even when the view label for the state is `<current>`, makes the view read-only.

White Box Test Testers check both the source code and the compiled executable file for problems that need to be fixed. The source code will have a view label to ensure that the testers are looking at an unchanging set of files. The view label will be attached to a promotion state named White Box Test. (White box testing is testing with full knowledge of what is in the source code.)

The executable files are not stored in the application because they can be easily built from the source code. Testers install them from a `Builds` folder on the network. This folder has child folders named `Build 1`, `Build 2`, and so on.

Change requests are entered against the executable files only. Developers make repairs in the current source code, sometimes reviewing the files with the view label attached to the Black Box Test promotion state.

Black Box Test Testers install the executable file, just as they would with white box testing. However, they do not need to see the source code or use promotion states with it. (Black box testing is testing with no knowledge of what is in the source code.)

Change requests are entered against the executable files only. Developers make repairs in the current source code, sometimes reviewing the files with the view label attached to the Black Box Test promotion state.

Alpha Test End users of the software product being developed install the product executable files and test the product in their own environments.

Change requests are entered by the alpha coordinator and/or the users against the executable files only. Developers make repairs in the current source code, sometimes reviewing the files with the view label attached to the Alpha promotion state.

Beta Test Beta testing is similar to alpha testing, but the group of users is greatly expanded because the product is much more stable.

Change requests are entered by the beta coordinator and/or the users against the executable file only. Developers make repairs in the current source code, sometimes reviewing the files with the view label attached to the Beta promotion state.

Release The product is now sold in the marketplace. Users install the executable file and call product support. Product support enters change requests against the executable files only. Developers make repairs in the current source code, sometimes reviewing the files with the view label attached to the Release promotion state.

The fixes go into future product releases and service packs to releases already in the marketplace.

In this example, every time the source files are used to produce a build (set of executable files) for testing, a view label is applied to the files to identify them for future reference. It is convenient to use view labels such as **Build 1**, **Build 2**, and so on, so that it is clear which source code files were used to create which set of executable files.

Over time, the build or view label associated with a promotion state will change. For example, the Release state may initially be associated with `<current>`, rather than a view label, because no files are candidates for release and no appropriate view label has been created. When white box testers decide that the set of files that they have examined is ready for black box testing, the view label associated with the White Box Test promotion state will be moved to the Black Box Test promotion state, and so on.

If promotion states are used, developers and testers who look at source code do not need to know that view label **Build 120** is currently being checked by white box testers, that the executable files for **Build 117** are currently undergoing black box testing, and other details.

Related Concepts

[Proper Use of Views](#)

Related Procedures

[Managing Labels and Promotion States](#)

[Configuring Promotion States](#)

[Promoting View Labels](#)

Personal Options

The application enables you and your team members to set personal options that suit your individual work styles. These options apply to the currently logged-on user on a given workstation. You can also update the list of servers available to your workstation and update your user account on the server on which you are currently logged in. The [Personal Options](#) item on the [Tools](#) menu allows you to adjust the way the following elements work.

For detailed information on setting these options, click the links below under **Related Information**.

Workspace

The application's Personal Options feature allows you to select a variety of options that affect the way your workstation operates. The options from which you can choose include:

- ◆ Confirming deletions, moves and shares, and warnings.
- ◆ Displaying toolbars and custom tools on the project view window.
- ◆ Indicating the effects of folder and tab changes.
- ◆ Controlling file status information
- ◆ Refreshing data automatically.
- ◆ Restoring the workspace on startup.
- ◆ Setting report and output paths.
- ◆ Specifying the contents of the StarTeam.log file.

StarTeamMPX

Servers that use StarTeamMPX offer additional caching services and performance enhancements. But to take advantage of these benefits, StarTeamMPX must be enabled on your workstation so that any open project view can take advantage of StarTeamMPX.

The right end of the application status bar displays the current status of StarTeamMPX on your workstation. The words and icons on the status bar for StarTeamMPX are as follows:

- ◆ An icon with a yellow lightning bolt indicates that StarTeamMPX is available and enabled for the currently selected project view.
- ◆ An icon with a grey lightning bolt indicates that StarTeamMPX is available for the currently selected project view but that it has not been enabled.
- ◆ An icon displaying a small circle with a diagonal line through it indicates that StarTeamMPX was enabled for the currently selected project view but that the connection has been broken. The network may not be working properly.
- ◆ No icon indicates that StarTeamMPX is disabled for the currently selected project view.
- ◆ **Instant** indicates that StarTeamMPX's autorefresh is turned on.
- ◆ **Auto** indicates that your workstation's autorefresh is turned on, but that StarTeamMPX's autorefresh is either turned off or unavailable.
- ◆ **Manual** indicates that your workstation's auto-refresh is turned off and that StarTeamMPX's autorefresh is either turned off or unavailable.

File Options

The Personal Options feature allows you to choose a variety of options to customize the way you work with files. In a few cases (such as Marking Unlocked Files Read Only), your administrator's choices may override your preferences.

The application offers many file options, including the following

- ◆ Controlling the time stamp for checked-out files.
- ◆ Using file checksums to determine status.
- ◆ Making unlocked files read-only.
- ◆ Speeding up check-outs by allowing the server to send only the delta between the working file and the requested revision.
- ◆ Automatically checking in merged files.
- ◆ Selecting an editor other than Notepad.

Change Request Options

Use the Change Request options to specify the criteria that the application uses to determine whether a change request has been read. You can also indicate how often the application should search for new change requests and how change request locking issues should be handled.

Requirement Options

Use the Requirement options to specify the criteria that the application uses to determine whether a requirement has been read. You can also indicate how often the application should search for new requirements and how requirement locking issues should be handled.

Task Options

Use the Task options to specify the criteria that the application uses to determine whether a task has been read. You can also indicate how often the application should search for new tasks and how task locking issues should be handled.

Topic Options

Use the Topic options to specify the criteria that the application uses to determine whether a topic or response has been read. You can also indicate how often the application should search for new topics and responses and how topic locking issues should be handled.

Related Concepts

[About Source Control](#)

Related Procedures

[Customizing Personal Options](#)

[Controlling How File Status Information is Stored](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

Understanding Default and Alternate Working Folders

Make sure that everyone is logged off from the server and that the server is locked before you change the Default Working Folder. It is just as critical to perform these actions as it is when you change custom fields or do anything else that affects all users.

When a view is created, a default location is specified for its working folder. If you change the Default Working Folder, not only the path to the working folder but the path to each child folder in the view may be similarly modified – not just for you, but for everyone working with that view. Therefore, before making such changes, it is important to understand the relationship of the working folder to the StarTeam view.

- ◆ The Default Working Folder path is used by everyone sharing that view, unless they have specified an Alternate Working Folder path. Only change the Default Working Folder if you want to change the path for everyone who shares the view.
- ◆ The Alternate Working Folder path lets you specify a different location for your own working folder than the Default Working Folder. If you do not want to use the Default Working Folder path, specify an Alternate Working Folder path — do **NOT** change the Default Working Folder. If you specify an Alternate Working Folder path, it is used instead of the Default Working Folder path. The Default Working Folder path must point to a location that is physically discrete for each user, such as a drive on that user's workstation or a personal directory on a shared file server.

The working folder for the view's root folder has an absolute path (for example `C:\New Product`). The path used for the working folder of a child folder depends upon how the child folder was created and what changes have been made to the path since that time. Generally, the working folder for a child folder is relative to that of the view (that is, relative to the working folder used for the root folder). For example, suppose that the path to the view's working folder is `C:\New Product` and that the root folder has a child folder named `Online Help`. In this case, the path to the `Online Help` working folder would be `C:\New Product\Online Help`. When the path to the view's working folder changes, the path to the child's working folder changes automatically.

If a new child folder is added to the view after it is created, the path to the child's working folder will usually be relative. However, if its working folder is on a different drive than the working folder for the root, its path will be absolute.

Related Concepts

[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Changing a View's Default and Alternate Working Folders](#)
[Managing Views](#)
[Creating and Configuring Views](#)

Check-in and Check-out Operations

One of the main functions of a source control management application is to place files into a project under version control. After it is under version control, team members can check files out, revise the , and check in a new revisions. The application preserves historical information about each file revision. And because of its linking capabilities, you can link a file revision to other items that affected the file or a particular revision of it. A number of other operations can be performed on files, such as moving a file or changing its branching behavior. This section specifically addresses the check-in and check-out operations.

In This Section

[Check-in and Check-out Overview](#)

Presents an overview of the check-in and check-out processes.

[EOL Conversion Handling Overview](#)

Explains how StarTeam handles EOL (end of line) characters for text files during conversion.

[Achieving Consistent Check-ins and Check-outs](#)

Describes how to achieve consistent check-ins and check-outs.

[Bulk Check-out Utility for Large Numbers of Files](#)

This topic describes the features provided by the Bulk Check-out utility.

Check-in and Check-out Overview

To place a file under version control, it must be added to a folder in a project view, which stores a copy of the file in the repository. After the file has been added to the repository, you and other members of your team can check it out, revise it, and check in new revisions, while the application maintains information on all revisions of the file. When checking out a file revision, you should verify that you have the tip or latest version of the file. Doing this ensures that the file you see contains the latest changes. If you intend to modify the file, you should check it out with an exclusive lock, to indicate to others that you are working on it.

If two team members change the same text file simultaneously, or if one member changes an outdated file, the application contains a merge option that allows the file changes to be combined so that no work is lost. In such cases, the application assigns a Merge status to the file. For details on file status states, see the “File Status Information” link below.

When you check in a file from a working folder on your computer, the application stores it by the MD5 value of its contents. If the file is identified as one that compresses well, it is compressed and placed in the hive's archive with a `.gz` extension. Otherwise, the uncompressed version is placed in the hive's archive. When you check out a file, the application copies the requested revision of that file to the appropriate working folder. If a copy of that file is already in the working folder, it is overwritten unless the working file appears to be more recent. In that case, you are prompted to confirm the check out.

You can perform check-out operations on more than one file at a time. For example, you can select files across multiple child folders using the All Descendants button, or you can check out all the files in the selected folder and its descendant folders using the **Check Out All** item on the **File** or context menu. This selection is equivalent to selecting **All Descendants**, all the files in the upper pane, and **Check Out**. When you use the **Check Out All** command, a confirmation dialog appears, regardless of your personal options settings.

Check-in and Check-out Recommendations

Every time you check a file revision out, its contents are copied to a working folder. Checking out a revision also ensures that you have the tip or a specific revision to work on. For example, you may need a team member's most recent changes to a file, or you may have deleted the working file from your hard drive and now need another copy.

Below are some recommendations about using files that are under version control:

- ◆ To let other team members know that you intend to make changes to a file, change the lock status to exclusive as part of the check-out process.
- ◆ As part of the check-in process, you can notify others both that you are finished making your changes to the file and that it is available for them to check out by removing the lock status.
- ◆ If you intend to continue making changes to the file but still want to check it in for backup purposes, keep the file locked.
- ◆ If two team members change the same text file simultaneously or if one member changes an outdated file, you can use the merge option to combine the changes in these files so no work is lost. In such cases, the application gives the file a Merge status.
- ◆ To prevent yourself from changing a file that you have not locked, select the **Mark Unlocked Working Files Read-only** personal option. Then, if you check out a file that you have not locked, the working copy becomes read-only.

About the File List Display

The files that appear in the upper pane depend upon the following factors:

- ◆ The application folder you select from the folder hierarchy in the left pane.

- ◆ The filter you select from the **Filter** drop-down list box above the upper pane.
- ◆ The depth you specify with the **All Descendants** button or by selecting **All Descendants** from the **File** menu. When this button is not selected, the application displays information for the selected folder only. When this button is selected, the application displays information for the selected folder, its children, its children's children, and so on.
- ◆ The application folder you select from the folder hierarchy in the left pane.
- ◆ The sorting, rearranging, and querying you perform on the list. The application sorts files in alphanumeric order, regardless of case.
- ◆ The files you have excluded from the display by using the folder's current or inherited exclude list.

Note: StarTeam recognizes the operating systems where the case-sensitivity of file names is important and handles files accordingly.

Related Concepts

[Check-in and Check-out Operations](#)
[Achieving Consistent Check-ins and Check-outs](#)

Related Procedures

[Checking In Files](#)
[Checking Out Files](#)
[Checking Out Files in Batches](#)
[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Related Reference

[File Status Information](#)
[Effects of Status on Check-ins and Check-outs](#)

EOL Conversion Handling Overview

This overview explains how StarTeam handles EOL (end of line) characters for text files during conversion.

StarTeam provides support for fixed EOL conversion files. For example, files can be checked out in **LF** format on every platform, regardless of specific options. Also, **Update Status** works for all text files once EOL Format is defined, regardless of what EOL format was used when they were checked-out. For compatibility with older Clients, if check-out "EOL conversion" is not requested, and EOL Format is **Undefined**, files are still checked out with the EOL conversion with which they were added to the Server.

Note: The default for automatic EOL conversion for check-out operations is “checked” if the user does not have that option defined already. Users that upgrade to 2009 should check that option to be sure they have it set correctly given the new EOL Format changes.

EOL Format Property

The EOL Format Property displays as **EOL Character** in the Cross-Platform Client **Item** pane.

The EOL Format property is only meaningful for text files during the check-out operation.

By default, the SDK will compute the EOL Format under the following conditions:

- ◆ When a new text file is added or a new revision is checked in for a text file whose EOL Format is **Undefined**, the file's EOL convention matches the platform default, EOL Format is set to **Client Defined**. Otherwise, EOL Format is set to the convention found: **Fixed LF**, **Fixed CR**, or **Fixed CRLF**.
- ◆ The user can change EOL Format to any value (other than **Undefined**) at any time.
- ◆ Regardless of their EOL Format setting, text files added or checked in with a StarTeam 2009 Cross-Platform Client always use a canonical (**CRLF**) format in the vault.

The EOL Format Property can be manually set in the Cross-Platform Client in the **Add/Check-in** dialog boxes, and the **File Properties** dialog box. Cross-Platform Client “EOL conversion for add/check-in” options are not available.

EOL Property Values

The EOL property value is displayed as **EOL Character** in the Cross-Platform Client **Item** pane.

The EOL Format property can be set in the Cross-Platform Client in the **Add/Check-in** and **File Properties** dialog boxes. If selected in the **Add/Check-in** dialog box, StarTeam uses the settings specified in the **File Properties** dialog box.

The EOL Property values are:

- ◆ **Undefined** (null in the SDK): Used for files added before StarTeam 2009.
- ◆ **Client Defined**: Causes workstation default or per-checkout EOL conversion option to be used.
- ◆ **Fixed CR**, **Fixed LF**, and **Fixed CRLF**: Causes this EOL format to be used always; the workstation/check-out conversion option is ignored.

Achieving Consistent Check-ins and Check-outs

Developers can use various StarTeam features to allow or avoid conflicts with other developers on the same files (in the same view). In a low-contention environment, developers can check-out files without locks, modify them, “refresh” to identify and resolve merge conflicts, and then check in modified files. All check-ins in StarTeam are atomic. If more than one file is checked in as the result of a single transaction, for example in a change package from a View Compare/Merge session, the files and their associated process items, are updated in a single action. If for some reason, the check-in fails, none of the files are checked in, and the status of the associated process items is not updated. In general, you can achieve consistent check-ins and check-outs by doing the following:

- ◆ Exclusively lock files before checking them in or out and unlock files after a successfully checking them in or out; or
- ◆ Temporarily change your view configuration to a known “stable” point

In higher contention environments, developers may want more assurance of getting consistent sets of files during check-out, that is, avoiding files that are a partial set of someone else’s check-in. The easiest way to achieve this need is “by convention”. Each developer exclusively locks all files before checking them in, and unlocks them when they are “complete”, either at check-in or soon thereafter. Correspondingly, each developer exclusively locks all files before checking them out, and unlocks them when complete. If a developer cannot get all the locks at once, they are potentially about to interfere with another developer, so they unlock files they currently have locked, wait a bit (probably talk with the developer they are conflicting with), and then try again. One implication of this “by convention” approach is that a developer could be blocked while waiting for another developer to finish-up.

A more formal way to enforce consistent check-outs is to use “view configuration”. To ensure consistent check-outs without locks, a developer can temporarily change their view configuration to a known “stable” point. In some organizations, a nightly build process creates a view label when the server is not used or lightly used. To temporarily change the view configuration from the StarTeam client, select **View ▶ Select Configuration ▶ Labeled Configuration** and choose the latest build label. (Alternatively, choose a timestamp or promotion state.) The view will switch to show the item states at the selected time, and “consistent” check-outs can be performed from there.

Note: The view configuration change is only at the client – the underlying “real” view is not modified. Also, note that “rolled-back” views are read-only: they must be reset to the “current” configuration before new/modified files can be checked-in. An implication to be aware of with this approach is that the time to switch the configuration can take a few seconds to a few minutes on very large views.

Related Concepts

[Check-in and Check-out Operations](#)

[Links: Internal and External](#)

Related Procedures

[Creating and Configuring Views](#)

[Checking In Files](#)

[Checking Out Files](#)

[Linking Items Internally or Externally](#)

[Linking Specific Revisions](#)

[Reviewing Linked Change Requests](#)

[Checking Linked Files In and Out](#)

[Selecting Linked Files](#)

[Customizing Link Properties](#)

[Deleting Links](#)

[Linking Files to Process Items](#)

Related Reference

[Effects of Status on Check-ins and Check-outs](#)

[File Options \(Personal Options Dialog Box\)](#)

[File Status Information](#)

Bulk Check-out Utility for Large Numbers of Files

With the Bulk Check-out (BCO) utility, which operates with the Cross-Platform Client, you can quickly check out a large number of files from an application repository to your working folders.

This topic contains the following information:

- ◆ Keyword Expansion and BCO
- ◆ Tips and Best Practices for BCO

Keyword Expansion and BCO

Since StarTeam 2005 Release 2 (patch 2), the Bulk Check-out utility (BCO) supports keyword expansion with check-outs performed from a Cache Agent. In previous releases, BCO only performed keyword expansion for non-Cache Agent check-outs. Keyword expansion is only performed on non-binary files that match the keyword extension list in the project.

During Cache Agent check-outs, BCO fetches every file that it can from the Cache Agent, but if it finds a `Log`-style keyword (`Log`, `$Log[x]$`, `$LogUTC$`, and `$LogUTC[x]$`), it issues a warning for that file and does not expand the keyword. Such files are subsequently fetched from the Server.

BCO supports the `$History$` keyword for both Cache Agent and non-Cache Agent checkouts. Since BCO supports the keyword expansion with the `$History$` keyword for Cache Agent-based check-outs, Borland recommends the use of the `$History$` keyword over the `Log` keyword and its variants.

Tips and Best Practices for Using BCO

The following are tips and best practices for working with BCO:

- ◆ You may want to use this utility to check out files for builds, as it is faster than the `stcmd co` option, the standard application command-line check-out, especially when combined with Cache Agent.
- ◆ You can create working folders with BCO using the `-cwf` option even if you are not checking out any files.
- ◆ Most build processes check-out from a view label or promotion state (using `-cagl` and `-cagp`). If you check-out tip revisions, be aware that users may be checking in the files that you are checking out, causing an inconsistent snapshot. You must use other means to ensure a consistent snapshot, if you need one. For example, you can exclusively lock the server using the `stcmd server-mode` command or ask users not to check in files until further notice.
- ◆ BCO does not recognize differences in folder names due to case-sensitivity. It does recognize these differences in file names.
- ◆ BCO returns the same error codes as the StarTeam command line utility (`stcmd`). Namely, value `0` is returned for success, `1` is returned for connection or other major errors, `101` is returned if some filename patterns were not matched, and `102` is returned if no filename patterns were matched.
- ◆ When checking-out files from StarTeam Server, BCO supports large files (> 2GB).
- ◆ When using the `$History$` keyword, StarTeam inserts a line containing only the `$` symbol after the last history entry. If you delete this line, StarTeam ignores the `$History$` keyword.

Related Concepts

[Achieving Consistent Check-ins and Check-outs](#)

Related Procedures

[Checking Out Files in Batches](#)

[Enabling Keyword Expansion](#)

Related Reference

[Table of StarTeam Keywords](#)

[File Status Information](#)

[Effects of Status on Check-ins and Check-outs](#)

Labels

This section discusses the use of labels.

In This Section

[Labels](#)

Describes revision and view labels.

[View Labels](#)

This topic describes best practices for using view labels.

Labels

In version control, the term *label* corresponds to the act of attaching a view or revision label (name) to one or more folders and/or items. StarTeam enables you to create two types of labels:

- ◆ **View labels** are automatically and immediately attached to all folders and items in a view at the time you create the view label. View labels have multiple purposes, but you primarily use them to place a *time stamp* on the entire contents of the view and as *build labels*. When you roll back the view to that label, you see everything that existed at that point in time—unless the label has been adjusted. You can create a view label for a specific point in time or as a copy of another existing view label. Unless the view label is frozen, you can adjust it to include or exclude some folders and items by attaching or detaching view labels. You can also move a view label from one revision to another.
- ◆ **Revision labels** are not attached automatically to any item in the view. Instead, they are used to designate a set of folders or items within a view. For example, you might want to label a group of files that should be checked in and out together.

About Labels in General

You can attach a label to any type of StarTeam item, including folder, files, requirements, change requests, tasks, topics, and audit entries. Any item can have more than one label. However, no two revisions of the same item can have the same label at the same time.

Every label is unique within its view. That is, no view label can have the same name as any other view label, no revision label can have the same name as any other revision label, and no view label and revision label can have the same name.

You can manually attach or detach both view labels and revision labels to or from a folder or item. In addition, you can use either type of label to identify a file when it is checked out. When you check a file in, you can attach and create a revision label for that file or attach an existing revision label.

You can select any type of item by its label. For example, you can select all files with a particular revision label and roll them back to that label, making the revision with that label the tip revision. Then you can compare your working files to the rolled-back revisions.

You can set access rights for labels at the view level or at the folder or item level. You must grant the rights to create labels, edit their properties, and delete them at the view level. However, you can grant the right to move a label (also called *adjust a label*) at the folder or item level.

Time Stamps and Build Labels

Using a view label as a *time stamp*, you can roll a view back to see everything in the view as it was at the time the label was attached. For example, to see if a particular file was in the beta version of a product, you can roll back the view to the beta label.

You may also use a view label as a *build label*, which allows the QA team to immediately determine what build to test for a fix to any given change request. To use a view label for this purpose:

- ◆ It must be designated as a build label.
- ◆ It must be created while the **Addressed in build** property for the change request contains the value Next Build.

When StarTeam creates the label, each change request with Next Build as its **Addressed in build** property will be reset to the build label.

To create a view label, you must select the current configuration of the view. Historical configurations are read only, and adding a label is considered a change. However, if a label already exists for a prior configuration, you can adjust

its name, files and folders can be added to it or detached from it, and so on. You can also move a view label from one revision to another.

For example, suppose your administrator creates a view label before each build, giving that label to the tip revision of each file in the view, and then checks out all the files with that label for the build. If the tip revision of one file does not change for a few weeks (or longer), it can acquire several view labels, while a file that changes frequently may have several revisions with no view labels and other revisions with only one view label.

When you detach a view label from a folder, StarTeam automatically detaches the label from everything in the subtree for which the folder is the root. If you roll back a view to a specific view label and a folder does not have that label, you cannot see the children of that folder and their contents anyway.

You can only create a view label at the view level and only while the configuration is current. However, you can create a view label for the current configuration or for a time in the past. In either of these two cases, StarTeam attaches the new label to the tip revision of each folder, file, change request, task, or topic that belonged to the view at the specified time.

You can also create a view label as a copy of an existing view label or as a copy of the view label currently attached to a promotional state. In these two cases, StarTeam attaches the new label to exactly the same items and revisions as the existing view label.

You can check file revisions out using this label or roll back the view to this label and see all the items associated with that label. For example, if you create the view label *Build 100* as you make Build 100 of your product from a view, all the files in the view will have the label *Build 100*.

If some items should not be included, you can detach the label from those items individually. For example, if some files should not have that label, select the files then select **Labels** ► **Detach** from the File menu or context menu to detach that label. If the files that should not be included all belong to the same folder and are the only files in that folder, use the Labels command on the Folder menu. For example, if the help files were not checked in until after the view label was attached, you can move that label from the previous revisions of the help files to the newly checked-in help files.

Label Access Rights

You can set access rights that apply to labels at the view level and at the folder/item levels. You set the access rights that allow a user or group to create labels, edit their properties, and delete them at the view level. For example, if you can create a label, you can set its initial properties. However, if you do not have the right to edit label properties, you cannot later freeze or unfreeze that label.

You can attach labels to individual folders or items, detach from them, or move from one of their revisions to another. The access right to move a label is named **Adjust a label**. You can grant or deny these rights at the folder or item level.

Related Concepts

[Promotion States](#)

Related Procedures

[Managing Labels and Promotion States](#)

[Attaching Labels to Items](#)

[Attaching Labels to Folders](#)

[Copying View Labels](#)

[Detaching Labels from Items](#)

View Labels

View labels should be used to mark configurations of the entire view that match specific milestones such as a new build number, a point from which another view was created, a release candidate, and so forth. Consequently, you don't create or use view labels for specific check-ins. Instead, you create them when one or more changes have been committed and it's time to launch a new build, spawn a release view, or promote changes to another view. The following two scenarios illustrate good uses of view labels.

The following two scenarios illustrate good uses of view labels.

Scenario 1: Daily Builds

Whether your team prefers daily builds, nightly builds, hourly builds, or builds on demand, this approach creates new view labels where "Use as build label" is selected:

- ◆ To get the process going, first create a view label as of a specific timestamp and launch the build process. (Some users prefer to have the build process itself create the view label.)
- ◆ The build process then checks out files attached to that label and launches compiles, links, and other build tasks. If these tasks are successful, the same build process could also launch unit tests and other verification tasks.
- ◆ If all build tasks are successful, the view label could be marked as "frozen", which identifies it as a "good build".
- ◆ Conversely, if a build task fails, the build process could generate a report or notify someone via email. If you're early in the development activity, you might choose to just move forward, allowing the team to continue making changes in the view. The next build process will simply "try it again". When you're later in the development activity, you'll want to have the appropriate developers fix their errors, re-attach the new revisions they create to the same view label, and perform the build again. Only when you achieve a "good build" is the view label marked frozen.

The advantage of this approach is that it tends to ensure that the tip revisions in a view are generally buildable. This supports a growing software development practice known as *continuous integration*. The disadvantage of this approach is it may be difficult with large teams and environments with lots of changes. It can result in a lot of broken builds, finger pointing, and "nasty gram" emails.

Scenario 2: "Change" Builds

Instead of relying on tip revisions being generally stable and buildable, another approach is to create view labels that are attached to revisions that are carefully selected. The steps that take this path are outlined below:

- ◆ Assume you start with an existing "good build" view label. As with the previous scenario, this label would be flagged as a build label and probably frozen.
- ◆ Although many changes are occurring in the build, you want to select only specific changes as candidates for inclusion in the next "good build" label. To do this, ensure that the corresponding file revisions are attached to a revision label and that this label is **only** attached to the file revisions you're interested in.
- ◆ Start the next "good build" label by "cloning" the current label. In the CPC, select **View > Labels > View tab > New**. In the corresponding dialog, choose "Labeled configuration" and select the current "good build" label. Your new label will now be identical to the old label.
- ◆ Select the file revisions associated with the desired changes. In the File tab, select **Select > By Label** and choose the appropriate revision label.
- ◆ Now attach these revisions to the cloned view label. In the File tab, select **Labels > Attach**. In the corresponding dialog, select the cloned label in the upper label list. In the "Attach to items at" group box, select "Labeled configuration" and choose the same revision label you used in the previous step. This ensures that the correct

revision of each file is attached to the cloned label—otherwise the tip revision will be attached, which may be the wrong revision.

- ◆ Repeat the previous two steps if there are multiple revision labels representing file revisions that should be included in the new label.
- ◆ Now launch the appropriate build and test process for your new “good build” label. Mark the label frozen or reattach fix revisions and retry as in the previous scenario depending on whether the build/test process succeeds.

This approach allows you to tag view configurations as candidates for builds, promotes, etc. without relying on the tip revisions being stable. The disadvantage of this approach is that if your latest “good build” label is way behind the view’s tip configuration, the quality of more recent changes may not be known for a while (which goes against the premise of continuous integration.)

Because this approach employs label “cloning”, there is a caveat we should mention with respect to deleted items. Suppose an item is attached to a view label and then deleted from the view. As you’d expect, when you “time travel” by adjusting your view window back to the view label, the item “reappears” because it existed when it was attached to the label. Less obvious, however, is that when you clone the label, the item will also be attached to the new label because the new label is initially identically to the old label. If you don’t want items deleted in the tip configuration to be attached to a cloned label, just detach them from the cloned label.

Related Concepts

[Overview of Views](#)

[Understanding View Types](#)

[Proper Use of Views](#)

Related Procedures

[Managing Views](#)

[Creating and Configuring Views](#)

Related Reference

[View Type Options and Settings](#)

Branching Options

This section contains conceptual topics related to branching options.

In This Section

[Understanding Branching](#)

Presents an introductory overview on branching.

[Overview of Branching Options](#)

Presents a basic overview of branching options.

[Branching Behavior of Items](#)

Describes the branching behavior of folders and items.

[Effects on Change Requests When Branched, Moved, and Shared](#)

Describes the effects on change requests when they are branched, moved, or shared.

[Floating Items](#)

Explains what floating means and how a floating configuration affects an item's behavior.

Understanding Branching

A branching view is a view that permits branching – that is, the folders and other items in the view can separate from the corresponding items in the parent.

Branching views serve many purposes. For example, you can create a branching view to:

- ◆ Meet different needs from those of your main line of development. For example, you might create a maintenance release or a custom version of your product, branched from a prior commercial release.
- ◆ Start development on the next release of your product by using some or all of the files from the previous release.
- ◆ Keep an area of your project private until it is completed and tested. Then you can merge your changes into the main line of development when and where necessary.

Only folders, files, and change requests can branch, although not every folder or every item in a branching view must branch. Requirements, tasks, and topics never branch.

Until an item branches, the corresponding items in both views remain identical. After an item branches, they are no longer identical, and the revision number indicates the new branch. The only way to make the items identical again is to manually merge them by comparing and merging views. After branching occurs, StarTeam no longer sends updates to nor applies updates from the corresponding item in the parent view.

For reasons of safety, deletions made in the parent view are not propagated to the child view and vice versa. If you want to delete a folder or item from all related views, you have to delete it manually from each of those views.

Also, a move is considered a copy operation followed by a delete operation. Consequently, the view in which the move was made has one copy of a folder or item in the new location, while the related views have two copies of the folder or item, one in the original location and one in the new location.

Note: Branching a view negates all shares, not just the ones between parent and child views.

Related Concepts

[Overview of Branching Options](#)
[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Understanding Branching](#)
[Branching Behavior of Items](#)

Related Procedures

[Reviewing or Changing Branching Behavior](#)
[Managing Views](#)
[Creating and Configuring Views](#)
[Changing a View's Default and Alternate Working Folders](#)
[Copying View Labels](#)
[Deleting Views](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Overview of Branching Options

Branching occurs when an item in the child view changes if its behavior is set to **Branch On Change**. When an item branches, a separation occurs between the item and its corresponding item in the parent view. These separate items also begin to have different branch revision numbers.

When creating a branching view:

- ◆ If you select **Branch All**, the behavior of every item that is in the view at the time the view is created is set to **Branch On Change**.
- ◆ If you select **Branch None**, the behavior of every item in the view at the time the view is created is not set to **Branch On Change**. Changes to any item with a floating configuration can be propagated to the parent view.
- ◆ When you branch a view, any manual shares between items in the same view are not retained in the view's child view.

Note: Any item with a frozen or fixed configuration is read-only when its behavior is not set to **Branch On Change**. Read-only means that no data about this item within the view can be changed. For example, although you may be able to edit a file, you cannot check it in or change its properties.

As you add, move, share, and modify items, their behaviors can change. The next few sections explain what particular behavior settings mean.

Branching is Disabled

When the check box for **Branch On Change** is disabled the item cannot branch. One of the following is true:

- ◆ The item is original to the current view, not shared into it. In other words, it is the root item in its own reference tree.
- ◆ The item has already branched. (An item can branch only once per view.)

Branching Is Set to Branch On Change

When **Branch On Change** is both enabled and selected, branching occurs the next time the item changes. At that time, a separation occurs between the item in the new view and its corresponding item in the parent view. The item that becomes separated from its corresponding item in the parent view takes on the following behaviors:

- ◆ Its **Branch On Change** check box becomes disabled
- ◆ Its revision number's dot notation expands to include two more numbers.

Branching Is Not Set to Branch On Change

When the Branch On Change check box is enabled but cleared, branching does not occur when you change the item.

If the item's configuration floats, the change is propagated to the parent view.

If the item's configuration does not float, the item cannot be changed because the parent view cannot be updated. The item is treated as though it were read-only. For example, if the item is a file, you can edit it but you cannot check it in or change its properties.

Related Concepts

[Understanding Branching](#)
[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Reviewing or Changing Branching Behavior](#)
[Reviewing or Changing Branching Behavior](#)
[Managing Views](#)
[Creating and Configuring Views](#)
[Changing a View's Default and Alternate Working Folders](#)
[Copying View Labels](#)
[Deleting Views](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Branching Behavior of Items

Given the appropriate settings for folders, files, and change requests, you can branch these items in a child view—that is, you can separate these items from the corresponding items in the parent view.

Branching a folder does not branch its contents (child folders nor items.)

After an item branches, it receives a new revision number. For example, if a file's revision number (in dot notation) 1.13 before the file branches, it becomes 1.13.1.0 after branching. The next change to the file in the parent view will receive the revision number 1.14. The next change in the child becomes 1.13.1.1.

Below are the basic facts about branching behavior:

- ◆ Folders and change requests branch when their properties change.
- ◆ Files branch when either their contents or their properties change.
- ◆ Requirements, tasks, and topics can never branch.

Typical Branching Scenario

Suppose you are working on a product and a customer requests a special edition of the product with a few special features tailored specifically for that customer. To separate the current product's items from those for the special request, a branching view is created.

When items are branched, they are derived from other items that become their ancestors. Items may have several completely different revision histories with common ancestries. In the case of a text file, for example, the branched item can later be merged with the file from which it originated. For example, the development of a product for a new operating system may start with the existing files for the first operating system as its base.

History Affects Branching Behavior

Whether or not a folder, file, or change request has the ability to branch depends on its history. If you do not know the complete history, you should not assume that you know its behavior. For example,

- ◆ If a folder or item was in the parent view at the time the branching view was created, and if the branching view was created with **Branch All** as its branching option in the **New View Wizard**, the folder or item's branching behavior is initially enabled and the **Branch On Change** check box is selected in the **Folder Behavior** dialog box.
- ◆ If a folder or item was in the parent view at the time the branching view was created, and if the branching view was created with **Branch None** as its branching option, the folder or item's branching behavior is initially enabled and the **Branch On Change** check box is cleared. However, this behavior can be changed.
- ◆ If a folder or item is added to the branching view after the view is created, the folder or item's branching behavior is disabled. The **Branch On Change** check box is disabled and cleared in the **Folder Behavior** dialog box. However, if you share that folder or item, its branching behavior becomes enabled automatically in its new view.

Branching Behavior of Shared Items

The "branch on change" behavior of a shared item is specific to the folder it is in. If the folder is in a project that resides on a 2005 Release 2 server, the "branch on change" checkbox is unchecked by default for the shared file. However, if the project resides on a 2006 or 2008 server, then the "branch on change" checkbox is checked by default for the shared file.

Related Concepts

[Overview of Branching Options](#)

[Overview of Views](#)

[Understanding View Types](#)

[Proper Use of Views](#)

[Understanding Branching](#)

[Branching Behavior of Items](#)

Related Procedures

[Reviewing or Changing Branching Behavior](#)

[Managing Views](#)

[Creating and Configuring Views](#)

[Changing a View's Default and Alternate Working Folders](#)

[Copying View Labels](#)

[Deleting Views](#)

[Sharing Folders or Items](#)

[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Effects on Change Requests When Branched, Moved, and Shared

The workflow of a change request may be significantly affected when the change request is moved, merged, or branches:

- ◆ If the **Last Build Tested** and the **Addressed In Build** fields have build labels as their values (in other words, if these fields are not empty and do not contain the value **Next Build**) the altered change request retains those values. In the new view, these values can be changed, but only to the names of build labels that exist in that view.
- ◆ If the **Addressed In Build** field contains the value **Next Build** at the time of the operation, this value is replaced by the name of the next build label created in the original view, not the next build label created in the new view. This action occurs even if other alterations have been made to the change request in the new view.
- ◆ If the **Last Build Tested** and the **Addressed In Build** fields have no values at the time of the operation, their workflow is specific to the view in which they currently reside.

Note: If a change request branches, its workflow is affected by its values in the **Last Build Tested** and the **Addressed In Build** fields at the time it branches.

Related Concepts

[Understanding Branching](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Reviewing or Changing Branching Behavior](#)
[Assigning Change Requests](#)
[Creating Change Requests](#)
[Changing a View's Default and Alternate Working Folders](#)
[Copying View Labels](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

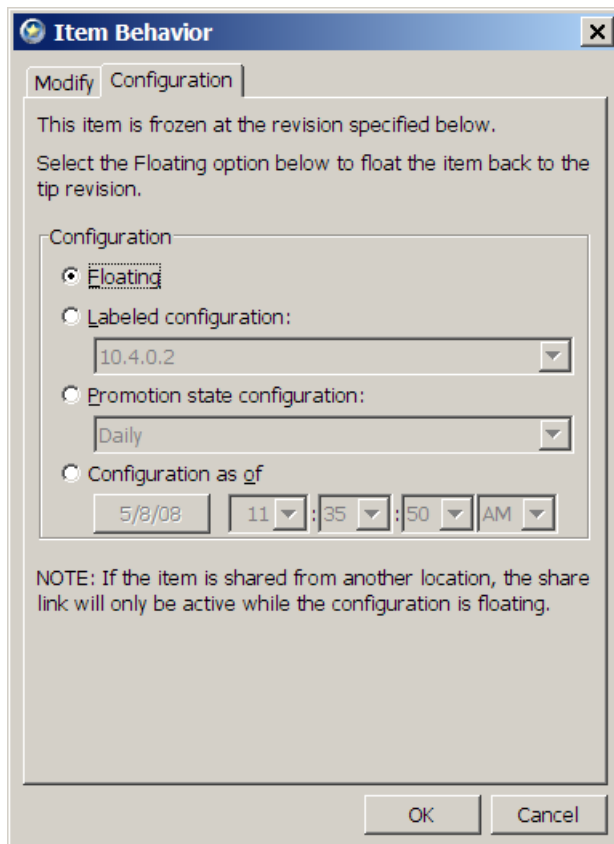
Floating Items

This topic explains what floating means and how a floating configuration affects an item's behavior.

About Floating Items

Floating is a StarTeam feature that needs careful management. Floating is a configuration option available for all items on the **Configuration** tab of the **Item Behavior** dialog box.

The alternative is a pinned configuration. Items can be pinned to a label, promotion state, or a point of time in the past. The StarTeam field named **Configuration Time** shows you the configuration of an item. The field is blank when the item is floating. The field contains a date and time if the item is pinned. That date and time may be the time that the view was created, the time the item was shared by a VCM operation, a time selected by a user on the **Configuration** tab in the **Item Behavior** dialog box, the time of the promotion state to which the item was pinned, or the time of the label to which the item was pinned.



Effects on an Item Caused by a Floating Configuration

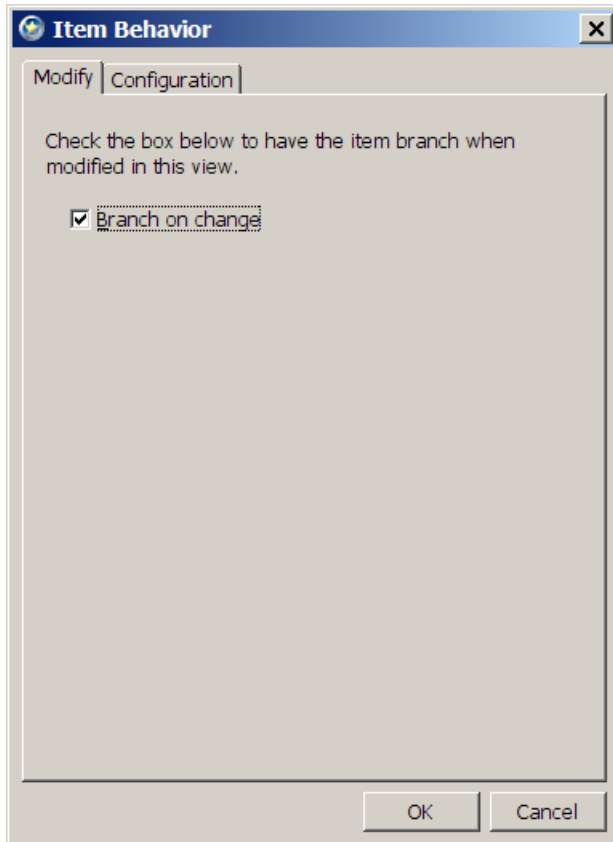
The effects on an item caused by a floating configuration depend on the item's behavior, for example if it is an item type that can branch: folder, file, or change request, and on the behavior of the folder in which it resides. The StarTeam field named **Branch State** shows you whether an item that can branch is a *Root*, *Branched*, or *Unbranched* item.

- ◆ A Root item is the root of a reference tree. It always has the dot notation 1.x.
- ◆ A Branched item is not the root reference in the reference tree and it has branched in the location where it resides. In its history, you should be able to see at least one place where its dot notation changed from 1.x to 1.x.y.0. Every time that an item branches two more digits are added to its dot notation. An item branches only

once in each location, but it may have been shared to many locations. The reference tree shows those locations.

- ◆ An Unbranched item is not the root reference in the reference tree, and it has not yet branched in the location where it resides. The StarTeam field name **Branch On Change** displays a **Yes** or **No** for unbranched items indicating whether the item is set to branch when it is changed. There is a check box on the Behavior dialog named **Branch on change** which is selected when an item is set to branch on change.

The following dialog shows that an item is set to branch on change. Its behavior can be changed from this dialog, but “Branch On Change” = yes is the preferred behavior:



If this item's **Branch State** had been Root or Branched, the check box would be disabled.

If this item's "Branch State" had been Root or Branched, the check box would be disabled.

Floating Shares

In general, floating shares are considered bad practice, especially for items that branch (folders, files, and change requests) because synchronization and propagation can happen automatically and may be unwanted.

- ◆ Changes from the parent item will automatically propagate to a child item if the child item's configuration is set to float. You can see this in the reference tree for either item. This behavior happens regardless of whether the **Branch On Change** option is selected or not. Once the child item branches, the floating stops. In order to branch, the item's behavior must be set to **Branch On Change**.
- ◆ Changes from the child item will automatically propagate to the parent only if the child item's configuration is set to float and the 'branch on change' option is not selected.

For floating folders, there are additional consequences:

- ◆ New items added to the parent folder can float into the child folder until the child folder branches.
- ◆ New items added to the child folder can float into the parent folder if the child folder's configuration is set to float and the 'branch on change' option is not selected.

Notice that floating can occur uni-directionally (parent reference to child reference) or bi-directionally (parent to child and child to parent). Also notice that a floating folder affects what happens to items newly added to it. As a result of floating, the parent and child items can be identical. Actually, they are the same item because no branching has occurred.

While floating is generally considered a bad practice, this feature is allowed because it can be useful in certain circumstances, particularly for items that cannot branch (requirements, tasks, and topics). When items that cannot branch have a pinned (non-floating) configuration, they are also read-only.

Note: An item that can branch is read-only if it is both pinned and set to branch on change. This is because the item cannot branch and the change cannot float to another location.

Some customers put items that cannot branch in a floating folder so that they can float from one view to another. They also set each item's configuration to floating so that it can be changed from any view in which it can be seen. Those changes, in turn, can be seen in all views where the item is visible.

VCM has a few of ways of finding floating items. It looks for the following item difference types:

- ◆ Floating Child Share (Row 1210 in the Action Decision Table)
- ◆ Floating Child Share, Source On Root Branch (Promote) (Row 1230)
- ◆ Target Folder Has Floating Share In Source (Row 200)

All of these are discussed in the topic "Understanding VCM Difference Types". Difference types 1210 and 1230 only occur if the VCM option **Fix Floating Child Shares** is used in the session. VCM can correct for these differences. When difference type 200 occurs, it results in a **Fail** condition. To commit the VCM session, a manual change must be made to the floating folder in the target view, followed by a refresh of the view. Then the VCM session must be redone. It might be wise to check other folders in the problem folder's reference tree. VCM only detects the problem between the two views being compared in the VCM session. There may be other floating references.

Related Procedures

[Configuring the Branching Behavior of Shared Items](#)

References to Folders and Items

The topics in this section provide getting started information about how StarTeam manages references to folders and items.

In This Section

[References Overview](#)

Provides an overview about how StarTeam manages references.

[Understanding References](#)

Describes actions that cause StarTeam to create references, how to view references in StarTeam, and the initial reference created by StarTeam.

[References Created by Branching Views](#)

Provides an overview about how StarTeam manages references for folders and items when branching views.

[References Created by Adding Items to Views](#)

Provides an overview about how StarTeam manages references when you add items to a view.

[References Created by Manually Sharing Objects](#)

Provides an overview about how StarTeam presents the reference hierarchy when you manually share objects from one location to another.

[References Created by Moving Objects](#)

Provides an overview about how StarTeam presents the reference hierarchy when you move objects from one location to another.

References Overview

You can base a folder or item in one application location on another folder or item stored in a different location within the same server configuration. **References** indicate the relationships between an original folder or item and the others based on it. References can be used to decide whether the changes you have made to a folder or item in one location need to be applied elsewhere.

Found in the lower pane of the clients, the **Reference tab** shows the relationships between the selected item and other folders or items with which it is associated. A folder or item may be associated with more than one project, view, or parent folder in the same server configuration because of sharing or because a child view has been created. Each instance of the original folder or item has a reference. Item references (including folders) can be viewed on the Reference tab of the lower pane. You can also view folder references from the folder tree in the left pane by selecting **Advanced** ► **References** from the Folder Tree or context menu to display a dialog.

Related Concepts

[Branching Options](#)

[Understanding References](#)

[References Created by Branching Views](#)

[References Created by Adding Items to Views](#)

[References Created by Manually Sharing Objects](#)

[References Created by Moving Objects](#)

Related Procedures

[Displaying Location References](#)

Understanding References

StarTeam creates at least one reference to a folder or item whenever:

- ◆ You create or add a folder or item to StarTeam.
- ◆ A branching child view is created that will contain that folder or item. As a branching child view is created from its parent, a subset of the folders or items in the parent becomes part of the child view. StarTeam automatically shares the folders and items in that subset into the child view.
- ◆ You manually share a folder or item from one location to another.

As you add, share, or move a folder or item, more than one reference to it may occur if the view is a child view that branches and floats, or if the view has child views that branch and float.

Actions Causing StarTeam to Create References

For example, suppose you want to move a file from one folder to another in the same view. Suppose that the view has two child views, both of which contain the file. That means that there are at least three references to this file, one in each of three views. Now, you move the file to another folder in the same view. The reference in the current view is moved to represent the new location of the file. Depending on the properties of the two child views, a new reference may be created for the file in each of the child views. The references in those child views to the file in its original location still exist, because the application does not assume that you want to change those references just because you have moved the file in the current view. You may end up with five references to this file that formerly had three references.

Note: Most administrators avoid branching, floating views if users are likely to perform many operations that result in additional references. For example, moving and sharing can result in multiple unwanted references to the same folders or items, which can cause confusion.

The following table explains what references StarTeam creates in the current view, the recipient view, the parent of the recipient view, and the children of the recipient view. This is often recursive. For example, if a reference is created in the parent view, new references might be created in the other children of that view or in the parent of that view, and so on, depending on what views are floating.





Actions Causing StarTeam to Create References

When a folder or item is...	...is a reference added to the view of the recipient?	...is a reference added to the parent view of the recipient view?	...is a reference added to the child views of the recipient view?
Part of a newly-created view	Yes, unless the new view is a reference view. In this case, a new view is not really being created, because a reference view is just a new way of looking at an existing view.) There is one reference for the folder or item in the newly-created view.	No, because the parent view is the source of the folder or item, so the reference in the parent view already exists.	No, because the newly-created view has no child views.
Added to the current view	Yes, there is one reference for the new folder or item in the current view.	Yes, if the current view is a branch none, floating child of the parent view. Otherwise, no.	Yes, if the child view is a branching (either branch none or branch all), floating child of the current view. Otherwise, no.
Shared within the current view	Yes, a new reference is created for the shared folder or	Yes, if the current view is a branch none, floating child of the parent view.	Yes, if the child view is a branching (either branch none


	item in the new location in the current view.	Otherwise, no.	or branch all), floating child of the current view. Otherwise, no.
Moved within the current view	No, the original reference is updated to reflect the move.	Yes, if the current view is a branch none, floating child of the parent view. Otherwise, no.	Yes, if the child view is a branching (either branch none or branch all), floating child of the current view. Otherwise, no.

How to View References

Consider the following example of four references that would display in the **Folder References** dialog box:

- ◆  **Help Files::Help Files::Help Files::starteam,1.0**
- ◆  **Help Files::Help Files\Freeze Check::Help Files\release 4\starteam, 1.0**
- ◆  **Help Files::Help Files\Freeze Check\New View::\Help Files\release 4\starteam, 1.0**
- ◆  **Help Files::Help files\variant 2::Help Files\release 4\starteam, 1.0.1.2**

In the above example, the selected folder has four references.

The Current icon  indicates which reference represents the currently selected folder or item. Otherwise, this dialog contains the same information regardless of the view in which you selected the folder.

Each reference shows the following, separated by double colons (::):






- ◆ The project name (for example, Help Files).
- ◆ The path from the root view to the view containing the folder (or item). For example, *Help Files\Freeze Check\New View*, where *Help Files* is the name of the root view, *Freeze Check* is a child of the root view, and *New View* is a child of the *Freeze Check* view.
- ◆ The path to the folder within the view. In the case of an item, the path is to the parent folder of the item.
- ◆ In the case of an item, the name or number associated with that item. This can be the filename, change request number, the requirement number, the task number, or topic number.
- ◆ The tip revision number for the folder (or item) in that view. (This information is separated from the rest of the reference by a comma, rather than the double colon.) For example, the folder in the example is revision *1.0* in all views except for the *variant 2* view (see the last leaf in the example tree). In the *variant 2* view, the revision number for the folder is *1.0.1.2* which indicates that the folder has been branched from the *1.0* revision in its parent view and has had three revisions in the *variant 2* view. Those revisions are *1.0.1.0*, *1.0.1.1*, and *1.0.1.2*.

In this example, the name of the project, the name of the root view, and the root folder in the root view all have the same name.

You can resize the **Folder References** dialog box (by dragging an edge or corner). It displays scroll bars when appropriate. The references in bold indicate which revisions of the currently selected folder or item are its descendants. In other words, the currently selected folder or item is part of the revision history for the references that are in bold.

Consider the following example from the Reference tab shows the references for a file ([AUDITSCC.DOC](#)). The reference for the currently selected file indicates that revision of the file is *1.6*. As indicated by the bolding of its reference, revision *1.8* is the only descendant of revision *1.6*. If a defect is found in revision *1.6* of

[AUDITSCC.DOC](#), the bolding helps you determine which descendants of 1.6 may also need the corrected lines. In this case, you may only need to update 1.8.

- ◆  **Help Files::Help Files::Help Files\starteam::AUDITSCC.DOC, 1.8**
- ◆  Help Files::Help Files\Freeze Check::Help Files\starteam::AUDITSCC.DOC, 1.1
- ◆  Help Files::Help Files\Freeze Check::New View2::starteam::AUDITSCC.DOC, 1.1.1.0
- ◆  **Help Files::Help Files\varc::Help Files\starteam::AUDITSCC.DOC, 1.6**
- ◆  Help Files::Help Files\variant 2::starteam::AUDITSCC.DOC, 1.2

Initial References

When you add a folder or item to the application, StarTeam creates a reference. Consider the following example of a folder hierarchy for a newly-created project. In this example, it is the folder hierarchy for the root view of that project.

Big Product
...Marketing Documents
...Online Help
...Source Code

Before you make any changes to the folder properties of the *Source Code* folder, the **Folder References** dialog box would contain exactly one reference to it. For example, **Big Product::Big Product::Big Product\Source Code, 1.0**.

As you make changes to the folder properties of the *Source Code* folder, the revision number might change from 1.0 to 1.1 and later 1.2. However, there will still be only one reference to this folder in the **Folder Reference** dialog box.

If a reference view is created (to be used, for example, by a group of reviewers), the view hierarchy for the *Big Product* project would contain two views, but the *Source Code* folder would continue to have just one reference. A Reference view contains a subset of the folders in its parent view, but those folders are the same folders as those in the parent view. They cannot branch. For example, after creating a reference view for reviewers, the Folder References dialog box would contain the following information:

Big Product::Big Product::Big Product\Source Code, 1.2.

Once you share the folder manually or share it automatically when you create a branching child view, additional references then display in the **Folder References** dialog box.

Tip: Do not confuse reference views with folder and item references. A reference view looks like a new view, but it is really a subset of an existing view. A folder or item reference is like a reference count. It indicates how many copies of the object exist or can exist if the object branches in each of its new locations. The creation of a reference view does not result in the creation of any folder or item references.

Related Concepts

[Branching Options](#)

[References Created by Branching Views](#)

[References Created by Adding Items to Views](#)

[References Created by Manually Sharing Objects](#)

[References Created by Moving Objects](#)

Related Procedures

[Displaying Location References](#)

References Created by Branching Views

When you create a branching view in StarTeam, each folder or item automatically shared from the parent view to the child view acquires an additional reference. In the view hierarchy (which you can display from [View](#) ► [Select View](#)), the new reference is a child of the original reference.

Folder References Created by Branching Views

Suppose that when the 1.0 version of Big Product ships, the team leader creates a branching view (based on the ship date for the 1.0 version) to be used for service packs, while new development on version 2.0 still continues in the project root view. These actions would result in the following view hierarchy:



- ◆ Big Product
- ◆ Big Product 1.0 Plus Service Packs
- ◆ Reference view for reviewers

At this point, two references display in the **Folder References** dialog box. When you are in the root view, *Big Product*, the **Folder References** dialog box for the *Source Code* folder contains the following information:



- ◆  **Big Product::Big Product::Big Product\Source Code, 1.2**
- ◆  **Big Product::Big Product\Big Product 1.0 Plus Service Packs::Big Product\Source Code, 1.2**

When you are in the child view, Big Product 1.0 Plus Service Packs, the Folder References dialog box for the Source Code folder contains the following information:



- ◆  **Big Product::Big Product::Big Product\Source Code, 1.2**
- ◆  **Big Product::Big Product\Big Product 1.0 Plus Service Packs::Big Product\Source Code, 1.2**

The Current (You Are Here)  icon indicates which reference represents the currently selected folder or item. Otherwise, this dialog contains the same information regardless of the view in which you selected the folder. StarTeam indents the reference for a child view beneath the reference for its parent. The references in bold indicate which revisions of the folder or item are descendants of the folder or item with the Current (You Are Here)  icon. In other words, the current folder or item is part of the history for the revisions that are in bold.

In the previous two examples, both references were represented in bold text. In the next example, this is not the case. This is because the properties of the *Source Code* folders in both the parent view and the child view have changed. The folder for the parent has revision 1.3, and the folder for the child has revision 1.2.1.0. Both folder histories have gone in different directions.






- ◆  **Big Product::Big Product::Big Product\Source Code, 1.3**
- ◆  Big Product::BigProduct\Big Product 1.0 Plus Service Packs::Big Product\Source Code, 1.2.1.0

The current folder is a descendant of itself, so it is always represented in bold text. However, it has evolved from the parent folder, so it is no longer in the history of the current folder. Accordingly, the **Folder References** dialog box for the parent folder would present the following information:

- ◆  Big Product::Big Product::Big Product\Source Code, 1.3
- ◆  **Big Product::BigProduct\Big Product 1.0 Plus Service Packs::Big Product\Source Code, 1.2.1.0**

File References Created by Branching Views

When you look at the history of a folder or item, you see its ancestors, not its descendants. However, if you change the tip revision in one location and that revision is an ancestor of the tip revision in another location, you might also want to apply your change to the tip revision in the other location (the object of the first descendant). The way to tell if a revision has descendants is to look at its references. Consider the following example showing the references for a file ([AUDITSCC.DOC](#)):

- ◆  **Help Files::Help Files::Help Files\startemp::AUDITSCC.DOC, 1.8**
- ◆  Help Files::Help Files\Freeze Check::Help Files\startemp::AUDITSCC.DOC, 1.1
- ◆  Help Files::Help Files\Freeze Check::New View2::startemp::AUDITSCC.DOC, 1.1.1.0
- ◆  **Help Files::Help Files\varc::Help Files\startemp::AUDITSCC.DOC, 1.6**
- ◆  Help Files::Help Files\variant 2::startemp::AUDITSCC.DOC, 1.2

As the bold text indicates, if the current revision is 1.6, then 1.8 is its only descendant. This also means that you would find revision 1.6 in the history for 1.8.

If a defect is found in revision 1.6 of [AUDITSCC.DOC](#), the bold text helps you determine the descendants of 1.6 that may also need the corrected lines. In this case, 1.8 may need to be updated. The other references are for revisions of the file that:

- ◆ Have already diverged (branched) and may be quite different than the current file.
- ◆ Are ancestors of the current file and less likely to need a change. For example, they may be in views that are read-only or no longer in use. Whatever the reason for the gap, the ancestors might require far more work than the changes you are about to check in.

You should check for descendants before (and perhaps after) you create a new revision of a folder or item. Before the change becomes a new revision in the application, you can see the descendants. Afterwards, you may see what other references have the same revision number as the newly-changed folder or item. If they, too, have the new revision number, then they, too, already have the new change. For example, the file may be floating in other views.

Related Concepts

[Branching Options](#)
[References Created by Adding Items to Views](#)
[References Created by Manually Sharing Objects](#)
[References Created by Moving Objects](#)
[Understanding References](#)

Related Procedures

[Displaying Location References](#)

References Created by Adding Items to Views

The addition of a new folder or item to a parent or child view can result in one or two references, depending on the relationship between the two views.

If the child view is a branching, floating view, StarTeam creates a reference in each view when a new folder or item is added to the parent.

If the child view is a branching, floating view created using the Branch None option, StarTeam creates a reference in each view when a new folder or item is added to the child.

Floating Down in the View Hierarchy

When a view has a branching child view (whether created with the **Branch None** or **Branch All** option) and the child view is floating, any folder or item added to the parent view becomes visible in both views. The history of the folder or item indicates the view in which the object was created, and the reference hierarchy displays the reference that identifies the parent view as the parent reference.

For example, if a file you add a file to the parent view, its history in either view shows the name of the parent view—until the file branches in the child view.

The following table shows the history in the parent view for a file that was added to the parent view and floated downwards.



Tip: You can review historical item information using the History tab in the client.

View	Revision	Branch Revision
Big Product	2	1.1
Big Product	1	1.0

The following table shows the history in the child view for a file that was added to the parent view and floated downwards. The history of the file displays the name of the view from which the file was originally added to the application—until the file branches. Then it displays the name of the view in which the file branched.

View	Revision	Branch Revision
branch none floating	3	1.1.1.0
Big Product	2	1.1
Big Product	1	1.0

If you were to display the References tab for this file (marketshares.doc) after it has branched in the child view, you would see the following information:

- ◆  Big Product::Big Product::Big Product\Marketing Documents::marketshares.doc, 1.1
- ◆  **Big Product::Big Product\branch none floating::Big Product\Marketing Documents::marketshares.doc, 1.1.1.0**

Notice that the history clearly shows the parent view as *Big Product* before the file branches. The history and references for folders and items added to the parent view are similar to those for folders and items that were in the parent view at the time the child view was created.

Note: The name of the views in these examples makes the information easier to understand. You would probably never name a *view parent* or any other of the names shown these examples.

Floating Up in the View Hierarchy

When a view has a branching child view (created with the **Branch None** option) and the child view is floating, any folder or item added to the child view becomes visible in both views. This is not true of branching, floating child views that were created using the **Branch All** option.

The history of the folder or item indicates the view in which the object was created, but the reference hierarchy always displays the reference that identifies the parent view as the parent reference.



The following table shows the history in the parent view for a file that was added to a child view and floated upwards. Notice that, even though this is the history in the parent view, the history displays the name of the view from which the file was originally added to the application.

View	Revision	Branch Revision
branch none floating	3	1.2
branch none floating	2	1.1
branch none floating	1	1.0

The following table shows the history in the child view for a file that was added to the child view and floated upwards. The history of the file displays the name of the view from which the file was originally added to the application—until the file branches. Then it displays the name of the view in which the file branched. In this case, those two views just happen to be the same view.

View	Revision	Branch Revision
branch none floating	3	1.1.1.0
branch none floating	2	1.1
branch none floating	1	1.0

When you view the reference hierarchy for a file that floats upwards, you cannot tell that the file was added to the application from the branching child view, and you must investigate the history (using the History tab) of the file to determine where the file originated. For example, the Reference tab would contain the following reference hierarchy for the slant.doc file that floated upwards:

- ◆  Big Product::Big Product::Big Product\Source Code\Timeout::slant.doc, 1.2
- ◆  **Big Product::Big Product\branch none floating::Big Product\Source Code\Timeout::slant.doc, 1.1.1.0**

Floating Up and Down in the View Hierarchy

If the view hierarchy is deep (the root view has grandchildren, great-grandchildren, and so on), the use of branching, floating views can cause a great deal of confusion. For example, suppose you add a file to a grandchild of the root view. Further, suppose that this grandchild view was created using the **Branch None** option and that its parent (a child of the root view) was created using the **Branch None** option. The file you add can float up to the parent and grandparent of the current view from which it will, in turn, float back down to the current view. This results in:

- ◆ One reference to the file in the current view
- ◆ One reference to the file in the parent of the current view (the result of floating up from the current view)
- ◆ One reference to the file in the root view (the result of floating up from the parent of the current view)

More references are created if the current view has floating children, grandchildren, and so on. Still more are created if the root view or parent view have other floating children besides the ones mentioned above.

Related Concepts

[Branching Options](#)

[Understanding References](#)

[References Created by Branching Views](#)

[References Created by Manually Sharing Objects](#)

[References Created by Moving Objects](#)

Related Procedures

[Displaying Location References](#)

References Created by Manually Sharing Objects

As you share a folder or item from one location to another (whether in the same view or a different one) an additional reference is created for that object in the new location. The reference for the new folder or item becomes a child of the reference from the folder or item that was shared.

Reference Hierarchy Example for a Manually Shared File

The following example shows two references for a file named *timeout.cpp*. The file was manually shared from a folder named *Source Code* to a folder named *Timeout* in the same view. Notice that the second reference is based on the first, but created as a by-product of creating a branching view.

- ◆  **Big Product::Big Product::Big Product\Source Code::timeout.cpp, 1.0**
- ◆  **Big Product::Big Product::Big Product\Source Code::Timeout::timeout.cpp, 1.0**

The application does not differentiate between references based on what caused them to be created. However, you can tell from the hierarchy that the first reference is the source of the second reference, because the second reference is indented under the first. You can also tell, because they are in the same view, that a manual share or move occurred. (The second reference would be in a different view if it was created automatically when a child view was created.

A shared folder or item can branch, but may never do so. Regardless, some subset of its history is part of the history of the original folder or item.





Floating Up and Down in the View Hierarchy



If the view hierarchy is deep (the root view has grandchildren, great-grandchildren, and so on), the use of branching, floating views can cause a great deal of confusion. Suppose all the views except the root view branch and float. At its new location, depending on how views were created, the folder or item you share can float:

- ◆ Up the view hierarchy from the recipient view to the root view
- ◆ Down to all the recipient children of the view, grandchildren, and so on
- ◆ From the recipient view's parents, grandparents, and so on, to all of their other children







This can result in a reference to the folder or item in the new location in every view in the project's view hierarchy. Many of those views may have already had a reference to the folder or item in its old location.

The following example shows all the references created by sharing a file named *shared within child view.doc* from one location in the *branch none floating* view to another location in that same view. The first three references are the references that existed prior to the sharing operation. The fourth reference is the new reference in the root folder. It is shown as a child of the first location in the *branch none floating* view because it floated up from that view. The fifth and sixth references resulted from references that floated down to the *branch none floating* child view of that view.

- ◆  **Big Product::Big Product::Big Product\Online Help::shared within child view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating::Big Product\Online Help::shared within child view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating\branch none floating 2::Big Product\Online Help::shared within child view.doc, 1.0**
- ◆  **Big Product::Big Product::Big Product\Source Code::shared within child view.doc, 1.0**

- ◆  **Big Product::Big Product\branch none floating::Big Product\Source Code::shared within child view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating\branch none floating 2::Big Product\Source Code::shared within child view, 1.0**

The next example shows that the file named *shareall.doc* existed only in the *branch all floating* view before it was shared to another view. The reference to the root folder starts the references that occurred as a result of the share operation. However, the recipient view could have been any of the other views, because the file would float up to the root and back down. On the way down, a second reference was created in the *branch all floating* view.

- ◆  **Big Product::Big Product::branch all floating::Big Product\Marketing Documents::shareall.doc, 1.0**
- ◆  **Big Product::Big Product::Big Product::shareall.doc, 1.0**
- ◆  **Big Product::Big Product\branch all floating::Big Product::shareall.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating::Big Product::shareall.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating\branch none floating2::Big Product::shareall.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating\branch none floating 2\branch none floating3::Big Product::shareall.doc, 1.0**

Related Concepts

[Branching Options](#)
[Understanding References](#)
[References Created by Branching Views](#)
[References Created by Adding Items to Views](#)
[References Created by Moving Objects](#)

Related Procedures

[Displaying Location References](#)

References Created by Moving Objects

When you move a folder or item from one location to another within the same view, StarTeam deletes the object at the old location and reinstates it at the new location. However, there can be side effects in that view's parents and children if any of the views are floating. This is because the copy at the old location is not deleted except in the current view. The parent and child views may end up with two references (one to the old location and one to the new location) instead of one to the new location.

Reference Hierarchy Examples

Suppose you move the file named *timeout.doc* from the *Marketing Documentation* folder to the *Timeout* folder in a given view that has no branching child views.




The following two examples show the references for this file before and after the move. The number of references is the same; only the path to the file has changed. The file has been deleted from its original location and added to its new location.

Before the move:  **Big Product::Big Product::Big Product\Marketing Documents::timeout.doc, 1.0**

After the move:  **Big Product::Big Product::Big Product\Source Code\Timeout::timeout.doc, 1.0**

However, suppose this view has a child view that was created without cutting off the connection to the parent (in other words the child view is branched and floating). In the child view, if the moved file has not yet branched, it is not deleted from its old location because you might really still want it here. However, it is added to the new location because it is perceived as a change to the parent that should be reflected in the child.

Notice that the file has only one reference in the parent but that it has two in the child view.

- ◆  **Big Product::Big Product::Big Product\Source Code\Timeout::timeout.doc, 1.0**
- ◆  **Big Product::Big Product\branched floating::Big Product\Marketing Documents::timeout.doc, 1.0**
- ◆  **Big Product::Big Product\branched floating::Big Product\Source Code\Timeout::timeout.doc, 1.0**

Some StarTeam users sort items using folders. For example, they decide to create a series of folders in a view to classify change requests by criteria such as:

- ◆ Will definitely make the next release
- ◆ Are under consideration for the next release (time permitting)

These change requests are usually moved from the root folder to one of the sorting folders, or later rearranged and moved from one sorting folder to another. This is a convenience in the current view, but it can cause multiple references in a parent or child view. If the view hierarchy is deep, the current view's parents, grandparents, children, grandchildren, and so on may be affected. Users who use such systems usually create child views that do not float.

Floating Up and Down in the View Hierarchy










If the view hierarchy is deep (the root view has grandchildren, great-grandchildren and so on), the use of branching, floating views can cause a great deal of confusion. Suppose all the views except the root view branch and float. At its new location, the folder or item you move can float:

- ◆ Up the view hierarchy from the recipient view to the root view
- ◆ Down to all the children, grandchildren, and so on of the recipient view

- ◆ From the parents, grandparents, and so on of the recipient view to all of their other children

A move operation results in one fewer reference to the moved folder or item in the view from which it was moved, and one more reference to it in the recipient view.

The following example shows that a file named *move within parent view* was moved from one location in the root view to another location in that same view (which is why there is only one reference to it in that view). Originally, the file was referenced in five views. The move caused a new reference in all the child views of the root folder, giving each of them two references to the moved file (one reference in its original location and one in its new location).

- ◆  **Big Product::Big Product::Big Product\Source Code::moved within parent view.doc, 1.0**
- ◆  **Big Product::Big Product\branch all floating::Big Product::moved within parent view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating::Big Product::moved within parent view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating\branch none floating2::Big Product::moved within parent view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating\branch none floating2\branch none floating3::Big Product::moved within parent view.doc, 1.0**
- ◆  **Big Product::Big Product\branch all floating::Big Product\Source Code::moved within parent view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating::Big Product\Source Code::moved within parent view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating\branch none floating2::Big Product\Source Code::moved within parent view.doc, 1.0**
- ◆  **Big Product::Big Product\branch none floating\branch none floating2\branch none floating3::Big Product\Source Code::moved within parent view.doc, 1.0**

Related Concepts

[Branching Options](#)
[Understanding References](#)
[References Created by Branching Views](#)
[References Created by Adding Items to Views](#)
[References Created by Manually Sharing Objects](#)

Related Procedures

[Displaying Location References](#)

Process Items, Process Links, and Process Tasks

This section discusses the use of enhanced process items, enhanced process links, and process tasks.

In This Section

[Process Items Overview](#)

Introduces process items.

[Process Items and Process Links](#)

Explains process item links.

[Process Rules](#)

Describes how each file add or check-in operation within a project can be required to link to a specific process item..

[Process Tasks and Enhanced Process Links](#)

Introduces enhanced process links and process tasks.

[Active Process Items](#)

Explains active process items.

Process Items Overview

Modern development practices require increased control over the entire development process. StarTeam enables developers to follow a defined development process, one that ensures that all file content changes be linked to either a change request, requirement, or task. Items used in this way are known as *process items*.

Specifically, a process item is a change request, requirement or task that is specified by the user as the reason for making a given set of changes. Process items are supported by the **Add Files** and **Check In** dialog boxes. As a result, source code and content are modified only to meet clearly defined and approved objectives, as expressed in the process item.

Out-of-view Process Items

Historically, the StarTeam Client has supported the selection of a process item from only within the current view. This functionality is useful in many processes, but it does not support a process where change requests, tasks, or requirements live in a different view than the source code files.

To support out-of-view process, the StarTeam Client now enables you to choose a valid process item for file add or check-in operation from any view on the same server as the files being committed. You can choose an item selected in the **Items** pane as the active process item for the current view, an open view on the same server, or a different view on the same server.

Also, the **Active Process Item** toolbar button contains a drop-down list enabling you to select the active process item from any opened view.

Process Item Usage

Using process items enables you to link and track changes made to your files, even when you and other members of your development team are not required to use process rules.

Note: The StarTeam administrator can enforce the use of process items for a project by establishing process rules. You define process rules in the Project Properties dialog box. Process rules specify that process items must be used when checking in files, and they establish which type of items can be used as process items.

When process rules are enforced, you must link and pin all files you add or check in to a process item.

- ◆ In the standard linking model, this action pins one end of the link to a special-purpose task known as a process item, and the other end to the new file or revision.
- ◆ In the enhanced linking model, this action pins one end of the link to the process item, and the other end to a special-purpose task known as a process task that is automatically created and maintained by StarTeam. The process task is then linked to the new file or revision. Note that if you use a task as a process item, the enhanced linking model still uses a second task in the role of process task.

If process rules are not enforced, you can still take advantage of the linking and tracking made possible with process items. As you add files or check them in, you indicate that the new file revisions are to be linked and pinned to a specific process item. You do this by selecting a change request, requirement, or a task as the process item for the operation. At the same time you can mark the change request as fixed, the requirement as complete, or the task as finished.

Note: If there is an active process item available, the **Check In** dialog box automatically fills in the **Process Item** field.

Using process items enables you to clearly distinguish the following:

- ◆ Which file revisions are related to or fix a specific change request.
- ◆ Which file revisions are related to or complete a specific requirement.

- ◆ Which file revisions are related to or finish a specific task.

Each view can have a different active process item. As you change from view to view, you will see the process item information displayed on the status bar change.

Process Item Behavior

In the context of a file add or check-in operation in a given view, a process item behaves according to either the rules of the standard model, or to those of the enhanced model.

- ◆ A process item that behaves according to the standard model is referred to as a *standard process item*. Links are created from the process item to each file whose content changed as part of the activity. Later, you can see the changes that were made as part of the activity by examining the links of the relevant PI.
- ◆ A process item that behaves according the enhanced model relative to a given operation is referred to as an *enhanced process item*.

These two linking models are described in more detail later in this section.

Process Items and Process Tasks

Process items act like lightweight change containers. They provide traceability, allowing you to trace file changes to their purpose or context. They also provide a way to identify file revisions for a specific change request, task, or requirement so that, for example, you can attach those revisions to a view or revision label.

Specifically:

- ◆ The process item can be a change request, a task, or a requirement.
- ◆ The process item can live in any view. It does not necessarily have to reside in the same view where the changes are being performed.

When you use enhanced process links in a project,

- ◆ Each process item has, at most, one associated process task per view in an active state.
- ◆ A process task is view specific.
- ◆ A process task for a given file add or check-in operation is always created automatically. For example, if you select a change request as a process item, StarTeam automatically creates the associated task if necessary, and automatically attaches the process links to the task instead of to the change request.

Note: Process tasks are no longer used in VCM Sessions. A change package object is created by the VCM session to track all changes in a single VCM Session.

Process Item Selection

The StarTeam client currently allows a process item to be selected as the active process item, which results in that process item being used by default in the **File Add** and **File Checkin** dialog boxes. The **File Add** and **File Checkin** dialog boxes also allow you to change the active process item prior to adding or checking in the files.

Related Concepts

[Change Requests](#)

[Requirements](#)

[Tasks](#)

Related Procedures

[Working with Change Requests](#)

[Linking Items Internally or Externally](#)

[Reviewing Linked Change Requests](#)

[Setting Active Process Items](#)

[Finding Files Linked to Active Process Items](#)

[Establishing Process Rules for Projects](#)

Process Items and Process Links

A link is a relationship between two items. StarTeam can link any two items in the same repository, regardless of project or view. You can use links to form associations between items for any reason. For example, you can link a requirement to a design document, a change request to a duplicate instance of the same defect, a task to a design specification, and so on.

A process item is a change request, task, or requirement used as the context for a set of file content changes. Process items can be used in any view for file add or check-in operations, but some projects may require them. Furthermore, project properties can define which item types are valid process items and which statuses are valid for each Item type.

Link Endpoints

Each endpoint of a link points to a specific item. An endpoint always stores the initial revision for which the link is valid. An endpoint can be pinned to a specific revision or float to the latest revision referenced by the item. A link also has a comment property that can be used to describe the link, why it was updated, and so forth. Links are versioned, which means that if you look at a historic configuration of a view, you'll see the links as they existed in the historic snapshot.

Process Links: Standard and Enhanced

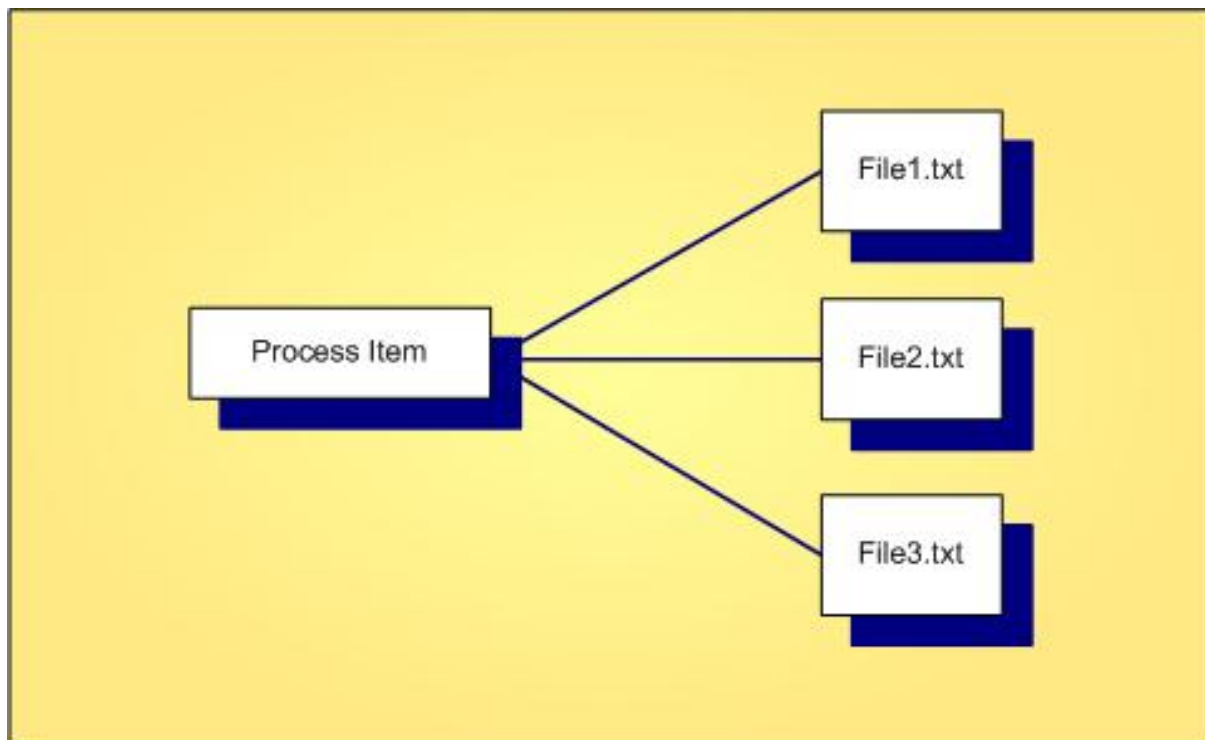
A *process link* is a StarTeam link that is used to indicate which revision of a file or a folder is relevant to a particular change. Process links are always pinned to a specific item revision at both ends. Links with floating endpoints are never considered process links.

StarTeam provides two models for implementing process links: The standard model and the enhanced model.

Standard Process Items and Process Links

StarTeam has historically created process links following the standard model. In this model, one end of each process link is to a process item. The other end is to a changed file.

For example,



When you add or check in files to a view in a project that uses standard process links, you select the process item to which your files will be linked. One process link will be created for each of the files in the operation. One end of the link will be pinned to a new file revision and the other end will be pinned to the current revision of the change request, requirement, or task.

As part of the add files or check-in process, you can change a change request's status to **Fixed**, a requirement's status to **Complete**, or a task's status to **Finished**. If the process item becomes fixed, complete, or finished, each new file revision is pinned to the new revision of the process item, the one with the correct status.

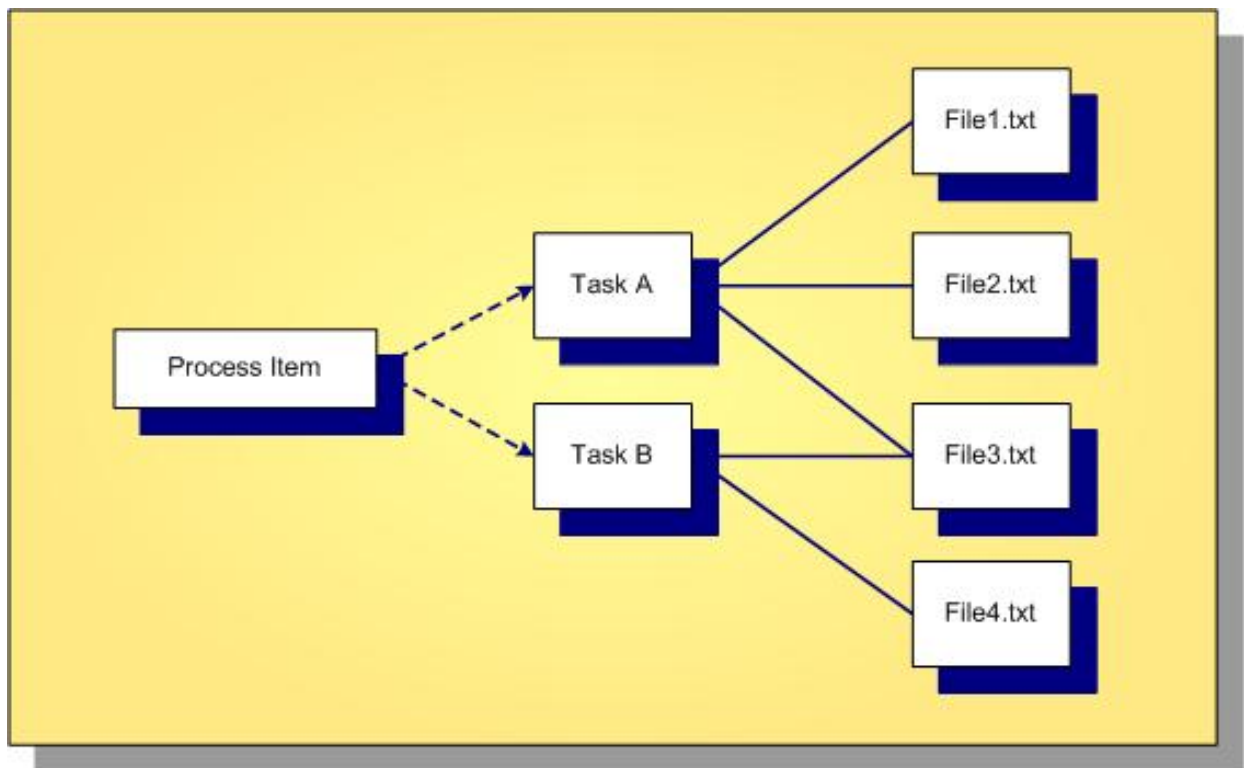
Note: If you have selected a disabled requirement as a process item, such as a read-only requirement published to StarTeam from CaliberRM, it cannot be marked complete.

When the application links a file to a process item through multiple add and check-in operations, it updates the existing link instead of creating a new link. Therefore, no matter how many revisions of the same file were pinned to revisions of the same process item, the latest revision of the file pinned to the process item is the only one specified in the link properties. To determine what file revisions are associated with a process item, you can review the links for that item.

Enhanced Process Items and Process Links

StarTeam now enables the use of enhanced process items and links. In the enhanced model, the process item (that is, the item specified as the reason for making a given set of changes) is distinguished from the task that represents the act of making the associated changes in a particular view. Changes are linked to the process item *indirectly*, through a process task.

For example,



In this model, the process item is linked to each relevant process task; each task holds the process links describing the changes for one file add or check-in operation.

In the enhanced model, one new process link is created for each new file revision created during the add or check-in operation. One end of the link is pinned to the tip revision of the process task, and pinned at the other end to the new file revision. In the enhanced model, StarTeam always creates a new process link rather than replacing an existing process link.

One advantage of the enhanced model is that StarTeam tracks every file revision related to the process item by using a separate link instead of continually updating a single link with each update to the same file. Another advantage is that a single process item can be used to track changes to files in multiple views even when different shares of the same file are modified. With the standard model, a link between a process item and a file in one view can have side-effects that invalidate links between the process item and a share of the same file in another view.

Note: In order to be interpreted as a process link, the link must be pinned at both endpoints. Furthermore, one of the endpoints must refer to a file or folder.

You can choose to use enhanced process links at the project level by checking **Use Enhanced Process Links** in the **Project Properties** on the **Process Rules** page.

A server administrator can also enable or disable the use of enhanced process links for all projects created on a specific server. When configuring a server, the server administrator has the option to specify one or both of the following:

- ◆ Enable and disable enhanced process links for all projects.
- ◆ Enable and disable enhanced process links for new projects.

External Linking

You can also use the external linking feature which provides the ability to link between items on different Servers (item to item linking across Servers). The process for creating external links is basically the same as for creating links between items on the same Server.

- ◆ All the same linking operations are available for external links that are available for standard links, such as Create Link, Complete Link, and drag and drop operations. To create external links, the projects on both Servers containing the items you want to link need to be opened in the Cross-Platform Client at the same time.
- ◆ External links are surfaced in the Links pane, where changes have been added to show the difference between internal and external links. For example, there is a new external link icon to differentiate external links from standard links. There is also a check box for viewing or hiding external links.
- ◆ The Link pane also gives you the ability to search for external links in all Servers to which you are logged on.
- ◆ When you create an external link, the item details for the external link on the **Link** pane are represented by a URL that contains the unique identity of that item.
- ◆ External link options are the same as standard links, such as being able to view the link properties. However, floating and pinning external links are not available.
- ◆ External links have direction, so whether you are viewing the external link from the source Item of the link, such as a CR, or the Item on the external Server, such as the file being linked to the CR, the source and target information will always remain the same and does not change based on the item you have selected.
- ◆ You can create external links only from items that exist in a StarTeam 2009 Server, and you must use the StarTeam 2009 Cross-Platform Client. Only the Source Item of the external link must reside in a StarTeam 2009 Server.
- ◆ You cannot create external links to or between Change Packages.
- ◆ Some information which is available for standard links is not available for external links because not all of the information from the external item is available, such as file status, who has the file locked on the external Server, what the item's folder path is, and the folder in which the item actually exists.
- ◆ **Access Rights** dialog boxes now contain **External Links** container level access rights.

Link Reports

StarTeam enables you to print reports of links associated with process items. You have the option of printing all links associating with a process item, or limiting the report to only enhanced links. Reporting is available from the **Reports** menu item for each type of process item.

Related Concepts

[Process Items Overview](#)
[Process Rules](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Reviewing Linked Change Requests](#)
[Setting Active Process Items](#)
[Finding Files Linked to Active Process Items](#)
[Establishing Process Rules for Projects](#)

Process Rules

Each project has the option of enforcing the use of process items by specifying certain *process rules*. When enforced, the process rules require you to specify a specific process item (change request, requirement, or task) for file add or check-in operations within the project.

Process Rules Review

You can determine whether process rules are in effect for a specific project – and what those rules are – by reviewing project properties. If you do not have the access rights necessary to do this, ask your administrator what process items apply to the project and what restrictions have been placed on them.

To set process rules, you must have the access rights required to change project properties. As a rule, only team leaders and administrators have these rights. To use process items, project users must have the necessary access rights, which are the rights to:

- ◆ See and modify the types of items used as process items in the project view.
- ◆ Create and modify links on files and process items.
- ◆ View and create tasks, and link to tasks if using the enhanced model.

Advantages of Process Rules

Establishing a system of process rules allows you to:

- ◆ Require that process items be used every time files are added or checked into the project.
- ◆ Stipulate that only certain types of items with specific statuses can be used as process items in the project.

If process rules are not enforced, linking and pinning to a process item during file add and check-in is optional, and you can select any change request, requirement, or task as a process item, regardless of its status.

Note: As a convenience, you can select a change request, requirement, or task as the *Active Process Item* prior to adding or checking in files so the check-in automatically use that process item during check-in.

Related Concepts

[Change Requests](#)

[Requirements](#)

[Tasks](#)

Related Procedures

[Working with Change Requests](#)

[Linking Items Internally or Externally](#)

[Reviewing Linked Change Requests](#)

[Setting Active Process Items](#)

[Finding Files Linked to Active Process Items](#)

[Establishing Process Rules for Projects](#)

[Promoting File Changes Into Baselines](#)

[Viewing Process Rules](#)

[Assigning Access Rights to Projects](#)

Related Reference

[Access Rights and Privileges](#)

Process Tasks and Enhanced Process Links

When you use *enhanced process links* in a project, the links for each file add or check-in operation in a given view are attached to a *process task*, rather than being attached directly to the process item. The process task and the process item are, in most cases, distinct. The process task is created automatically by StarTeam.

In the enhanced model, new process links are always added to a process task whose status is *In Progress* in the target view. If there is no process task in progress for a given file add or check-in operation, StarTeam creates one just before adding the first process link for that operation. A process task that is In Progress can itself be selected as an active process item.

Note: Process tasks are not used by the View Compare/Merge process. Instead, all changes in a VCM session are stored in a change package object which can be committed in the session, or saved for review and committed later.

Advantages of Enhanced Process Links and Process Tasks

By using enhanced links, StarTeam creates the process task to track the changes for a process (such as fixing a bug or finishing a task) rather than using the process item itself.

In the enhanced linking model, StarTeam items still represent process items for add and check-in operations, and process links identify the changed files and folders. However, the process item and the changes are represented by separate items.

Specifically,

- ◆ The process item can be a change request, task, or requirement.
- ◆ The process item can live in any view. It does not necessarily have to reside in the same view where the changes are being performed.
- ◆ A process task is view specific. It is always created in the root folder in the view where the changes have been made.
- ◆ The process task for a given process item is always created automatically and as transparently as possible. For example, if the you select a change request as a process item, StarTeam automatically creates the associated task if necessary, and automatically attaches the process links to the task instead of the change request.

This model makes it easier to see how a process item was used across various views in the project, no matter where the process item is referenced. It also makes it easier to support process items from another view.

Note: For projects that use the enhanced linking model, the **Link** tabbed pane gives you the option of displaying all links or only enhanced process links. In addition, the **Link** pane displays the current **Item Selection** and allows for navigating to child links.



How Process Tasks Work When Checking In Files

In the standard linking model, if you check in three files using a process item, the process item is linked and pinned to the tip revision of the files. With the enhanced linking model, you use process tasks for check-in and add-file operations, and the active process item is linked directly to a process task. That process task is linked and pinned to the file revisions. The process task serves as an intermediate item between the process item and the files. A process item is relevant to only one view. If you reuse a process item in a different view, StarTeam creates a new process task for it in that view. If you use a process item repeatedly with the same process task, StarTeam creates new links for each operation.

Distinguishing Process Tasks from Other Tasks

When you select an enhanced process item in the StarTeam client, the **Link** pane displays a list of all process tasks that are relevant to this process item, regardless of which view the corresponding operation occurred in. Also, the name of the relevant view is shown in each case.

StarTeam distinguishes a process task from a standard task in the **Link** pane by displaying a different icon for each type of task:

Task Type	Icon	Description
Standard Task		Created explicitly by the user for a specific task.
Process Task		Created automatically by StarTeam to track a set of changes in a given view.

You can also distinguish process tasks from standard tasks by looking at the value of the task's **Usage** property in the **Detail** pane.

Note: To make the **Usage** property visible in the **Detail** pane, you must select it as one of the **Fields** to show in the **Filters** ▶ **Show Fields** dialog box.

The following are the **Usage** property values used for the various types of tasks:

- ◆ For process tasks created when adding and checking in files, the **Usage** property value is **Checkin**.
- ◆ For tasks that are not process tasks, the **Usage** property value is **Other**.

Tip: You can filter process tasks from standard tasks using the **Usage** task property.

Related Concepts

[Change Requests](#)

Related Procedures

[Setting Active Process Items](#)

[Linking Items Internally or Externally](#)

[Reviewing Linked Change Requests](#)

[Setting Active Process Items](#)

[Finding Files Linked to Active Process Items](#)

[Establishing Process Rules for Projects](#)

Active Process Items

The StarTeam Client enables you to pre-select a process item as the active process item to use the next check-in operation. This type of process item is referred to as an *active process item*. You can select a change request, requirement, or a task as an active process item before adding or checking in files. Pre-selecting an active process item is a convenient way to save time when you know that you will be adding files or checking them in later using the designated process item. When you have a process item selected on the upper pane, making it the active process item is a simple operation. The active process item is the default selection when you add files or check them in; however, you can change your mind and select another appropriate item.

An active process item is used until its status becomes ineligible or another process item is chosen. Before or during each check-in operation, make sure you select the correct *active process item*.

The **Status Bar** displays the name of the active process item. If the active process item is from a different view and project than the current view, you can hover over the active process item shown in the status bar and see a tool tip showing the project and view as well as the process item description. The **Status Bar** pre-pends the name of the active process item with the name of the project and view. To see details about a process item, you can double-click the item in the status bar to open the Properties dialog box with more information.

Note: You can only specify one item for each view as an active process item. Selecting a second active process item clears the first.

Tip: After you finish with a process item, you should choose **Clear Active Process Item** on the menu so that it cannot be accidentally reused. That removes the information from the status bar and keeps the process item from reappearing in the **File Add** or **File Checkin** dialog boxes.

Related Concepts

[Change Requests](#)
[Requirements](#)
[Tasks](#)

Related Procedures

[Working with Change Requests](#)
[Linking Items Internally or Externally](#)
[Reviewing Linked Change Requests](#)
[Setting Active Process Items](#)
[Finding Files Linked to Active Process Items](#)
[Establishing Process Rules for Projects](#)
[Promoting File Changes Into Baselines](#)
[Viewing Process Rules](#)
[Assigning Access Rights to Projects](#)

Related Reference

[Access Rights and Privileges](#)

Comparing and Merging Files, Folders, and Views

This section contains conceptual information about merging files, folders, and views.

In This Section

[Overview of File Compare/Merge](#)

Gives an overview of File Compare/Merge.

[Overview of View Compare/Merge \(VCM\)](#)

Presents an overview of comparing and merging views.

[Change Packages Overview](#)

Describes the change package feature.

[Advantages of Change Packages](#)

Discusses the advantages of change packages.

[View Compare/Merge Session Perspectives](#)

Presents an overview of the three perspectives in a View Compare/Merge Session.

[View Merge Type Scenarios](#)

Presents some typical full-lifecycle scenarios using VCM for cross-view change management.

[VCM Merge Types, Rules, and Scenarios](#)

Presents an overview of the types of merges you can perform and the rules that apply to them.

[Access Rights Required for StarTeam VCM Users](#)

Describes which access rights VCM users need for projects, views, folders, files, and other StarTeam items.

[Tips for Successful VCM Sessions](#)

Presents some tips for successful View Compare/Merge sessions.

[View Compare/Merge Action Decisions](#)

Reference topics for View Compare/Merge session action decision.

Overview of File Compare/Merge

File Compare/Merge is a graphical file and folder comparison and merge tool delivered with StarTeam. It enables you to compare the contents of two files or folders, and manually or automatically merge the contents. The File Compare/Merge panes highlight differences using a configurable color scheme, and dynamic action buttons display in the highlighted areas to simplify the merging process.

There are three versions of File Compare/Merge, and how you start File Compare/Merge determines which features are available. The ability to edit text files in a File Compare/Merge pane depends on which version of File Compare/Merge you are using, and what type of files you are comparing and merging. The following list describes editing capabilities in each version of File Compare/Merge.

Main File Compare/Merge	Edit a local file, a merged base file, and a copy of a repository revision.
Embedded File Compare/Merge	No editing is possible.
Standalone File Compare/Merge	Edit all files being compared and merged.

Note: You cannot edit the actual historical revision of a file in the StarTeam repository.

Main File Compare/Merge

The main File Compare/Merge window is displayed using menu commands in the client, and it opens in a separate window enabling you to do the following:

- ◆ Compare the contents of a local file with the tip revision stored in the StarTeam repository. You can also edit the contents of the local file from within the File Compare/Merge window, and save the changes for check-in.
- ◆ Compare two revisions of a file listed on the **History** tab in the StarTeam application window. Editing the content of historical revisions is not allowed.
- ◆ Merge the contents of a local file with the tip revision in the StarTeam repository. The merge results are stored locally, and the file status is changed to **Modified** in StarTeam so you can check in the file.
- ◆ Edit the temporary local copy of a repository revision and save it as a file with a different name.

The main File Compare/Merge gives you the option of viewing a third pane for displaying the merge results. You can edit the contents in the third pane and save your merged results.

You can start File Compare/Merge using context menus on selected files in the StarTeam client, or by checking in an older version of a file which causes a merge situation. You can also access File Compare/Merge directly from the View Compare/Merge tool. For more information, see the topic “Comparing a Local File with a Repository File” in the links below.

Embedded File Compare/Merge

The embedded File Compare/Merge gives you a quick way to do a comparison of text in two files or versions of the same file, as well as compare properties of non-file items such as change requests. If you are comparing two text files, it performs a dynamic comparison of two selected files, or the selected repository file with your local working copy. It displays the text contents of both files in an embedded pair of panes at the bottom of the StarTeam window. These embedded panes only compare the contents of two text files, and do not permit editing or merging.

If you are comparing the properties of two non-file items, only the property values of each selected item display in the embedded panes.

When the embedded File Compare/Merge is activated, it immediately compares the files or properties when you do one of the following:

- ◆ Select a file or other item in the upper pane, or the **History** pane. The comparison is between the local working copy of the item and the selected item revision in StarTeam.
- ◆ Select two items in the **History** pane. The comparison is between the two historical revisions in StarTeam.
- ◆ Select a linked item in the **Link** pane. The comparison is between the selected item revision and its previous revision.

Standalone File Compare/Merge

The standalone File Compare/Merge is started outside of the StarTeam client from the Windows **Start** menu. This File Compare/Merge compares files, folders, and images, and it can merge the contents of two text files or two folders. Unlike the main File Compare/Merge, the standalone version does not compare any local files with files that are in the StarTeam repository. It compares and merges two or three files, folders, or images that are on your local computer or network. You can edit text file contents directly in the File Compare/Merge panes, and you can move lines or blocks of text between the panes. You can also move folders between the panes during a folder comparison or merge.

For more information, see the topic “Comparing Two Local Files” in the links below.

Related Concepts

[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Folders](#)
[Comparing Images](#)
[Merging a Local File with the Tip Revision](#)
[Merging Two Local Files](#)
[Merging Folders](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Overview of View Compare/Merge (VCM)

At key milestones in a development project, most views have changes that need to be propagated to another view, and sometimes to several views. When work is complete or reaches a suitable release point in an activity or sandbox view, it needs to be merged “up” to the parent view. When a bug is fixed in the main view, it may need to be propagated “down” to one or more release views. Sometimes a fix needs to be propagated “sideways”, from one release view to a sibling release view.

StarTeam provides you with a comprehensive cross-view change management tool for propagating discrete sets of changes from one view to another called View Compare/Merge (VCM).

VCM is available through two application interfaces:

- ◆ The Cross-platform Client (CPC) provides a graphical interface that is integrated with other CPC functionality. The graphical VCM tool provides interactive compare/merge functions with per-item and per-folder interaction. This allows you to carefully control which items are compared, and how each difference is resolved. For example, you can select an item, folder, or group of items in a regular view window and start a VCM session using the selection as the source scope.
- ◆ The [VCMUtility](#) is a command-line tool that allows you to start, resume, and commit VCM sessions from shell scripts. In a single script, for example, you could create a VCM session, perform a test build of the proposed changes and run other verification tasks, then commit the session if all tests complete OK.

Compared to the use of process items, which tracks file changes within a view, VCM is the preferred mechanism for propagating and tracking changes made across views, that is, from one view to another. A common use of VCM is to propagate the changes in one view, as tracked by a process item, to another view.

About VCM Sessions

VCM uses a “session” concept, which is a specific view compare/merge operation that can be started, reviewed, adjusted, verified, and then committed. When you save a VCM session, a *change package* object is created that contains all the changes and information about the changes made during that session. Except for locks, no data is modified in the StarTeam repository until the change package for that change session is committed, at which point all changes are applied at once to the target view.

Before a change package is committed, it can be saved and restored – potentially on a different machine. This allows one user to create the change package, then transfer it to a peer for review or to verify the change package before committing it.

The Cross-Platform Client interface allows user interaction with the entire VCM process. The [VCMUtility](#) provides partial or complete automation of VCM sessions, which can be helpful when a specific kind of session is performed repeatedly. VCM sessions are interoperable between the two interfaces. For example, you can create and save a change VCM session using the **VCMUtility**, then restore, review, and commit the session using the Cross-Platform Client. You can also apply the same changes in the committed change package to another view using a process called “Replay”.

VCM Session Views

Two StarTeam views are involved with every VCM change session: a *source view* and a *target view*.

- ◆ The source view contains the changes to be propagated. You can use the tip configuration of the source view or a snapshot defined by a view label, promotion state, or timestamp. There are several ways to identify the items to be “considered” for propagation, collectively known as the source scope. If you like, the entire view can be used as the source scope, though you will usually propagate items that represent specific changes.
- ◆ The target view receives the propagated changes, so it must be able to be updated. “Receiving” the changes does not mean that the source items are merely copied to the target view as-is. VCM compares the source view to the target view and determines if and how to propagate item differences. Propagation of a specific

change can take many forms such as sharing, re-pinning, merging, and so forth. You can review and adjust what VCM proposes before anything is committed.

Note: The source and target views must belong to the same project.

VCM Merge Types

Every VCM merge session has a specific *merge type* which controls the direction and semantics of the VCM session.

- ◆ **Rebase:** A rebase session propagates changes from a parent view (the source) to an immediate child view (the target). The most common use of rebase sessions is to “catch-up” a child view with changes that occurred in the parent since the child view was created or since the last rebase was performed.
- ◆ **Promote:** A promote session is the opposite of a rebase. It propagates changes from a child view to its immediate parent. Promote sessions employ special rules that accommodate the directionality of share trees. That is, they strive to keep share trees pointing in the same direction as the project’s views. (See the discussion for the Main View in section 5.3.2.)
- ◆ **Replicate:** A replicate session propagates changes from a source view to any other (updateable) view in the same project. Compared to rebase and promote, replicate is the “wild card” merge type that allows changes to be propagated anywhere in the same project.
- ◆ **Compare-only:** A compare-only session only shows you the differences between views or two configurations of the same view. No changes or merge actions are performed.

For more information on merge type rules, see “View Compare/Merge Types and Rules”.

Why Use VCM?

There are a variety of reasons you might want to merge two views. A typical example would be code development in a software development company. If both maintenance and new development for a software product need to occur in parallel, a separate view is often created to store each set of source code files. If you fix some defects in the maintenance view, you may need to merge changes from that view into the new development view so that the next release of your product has the fixes.

You can compare and merge any two views if those views are in the same project. However, in general, you can merge only items that are related (that is, have a common ancestor), even though that ancestor may not be in either of the views you are comparing or merging. The exception to this is when files are matched by name.

VCM allows you to do the following to reconcile differences between views:

- ◆ Catch up an activity or child view with new changes from its parent (*Rebase*).
- ◆ Deliver changes or a completed activity from a child view to its parent (*Promote*).
- ◆ Duplicate a fix or enhancement from one view to another view (*Replicate*).
- ◆ Merge items of one view, known as the source view, with items in a second view, called the target view. Changes are stored as updates to the target view.
- ◆ Merge two views using an earlier configuration of the source view.
- ◆ Compare and merge file contents and item properties.
- ◆ Control what is being merged by using one of several selection methods before starting the VCM session. For example, individual items may be selected, a folder and its contents may be selected (optionally recursively), or a process item and its linked files can be selected.

VCM always uses the source file (that is, the “merge from” file) from the last recorded merge as the common ancestor for performing three-way merges. When there is no recorded merge point, View Compare/Merge uses the most

recent common ancestor. For example, if the item in the source view has the dot notation 1.9 and the item in the target view is 1.7.1.4, the most recent common ancestor is 1.7.

Note: For more details on the different types of merges, see "View Compare/Merge Types and Rules" in the links below.

The VCM Process

View Compare/Merge provides consistent view compare/merge features across all client platforms. It supports refactoring and directionality. View Compare/Merge is optimized for distributed teams, and is oriented around merge tasks to facilitate the way you really work. It also allows the review and commit phases to be divided across time.

All VCM sessions follow the same basic process which is divided into the following phases: *Definition*, *Comparison*, *Review*, *Save and Restore*, and *Commit*.

Definition Phase

The VCM parameters are specified for the compare phase. The session is defined in terms of its merge type, the source view (and snapshot), the target view, the source scope, and various options.

- ◆ In the Cross-Platform Client, a VCM session is started from a "View Compare/Merge" menu option which is available in several places, and which launches the **View Compare/Merge Wizard**. The context from which the wizard is launched affects the definition of the session.
- ◆ In the `VCMUtility`, the session is defined by command-line arguments and/or an options file.
- ◆ Both the Cross-Platform Client and `VCMUtility` provide a variety of ways to define the source scope items. The source scope can be the file revisions linked to one or more process items (and optionally the process items themselves); the items attached to a revision label; items of specific types that reside in selected folders; a set of explicitly selected items; and items of selected types in the entire view.

The definition phase is where you define the process for the comparison. In this phase, you can:

- ◆ Specify the view merge type.
- ◆ Specify the source view, item(s), target view, and options.
- ◆ Specify the type of items to include.

Comparison Phase

The Comparison phase is where the comparison takes place. The comparison takes place automatically when you finish the wizard before the View Compare/Merge opens in the Cross-Platform Client.

Once a session starts, VCM first performs the Comparison phase.

- ◆ "In scope" source items are matched to corresponding items in the target view. VCM detects both simple changes (items that are new, modified, or deleted in one or both views) and more complex changes (items that have been moved, renamed, modified-and-moved, and so on.)
- ◆ For an item that exists in both the source and target view, VCM determines if either item or both have changed since the last time they were merged.
- ◆ For each difference found, VCM defines an *item difference* that specifies the source and/or target item (sometimes there is only one of these) as well as a default *action*. The action specifies what the session will do about the item difference. Many actions are possible including *ignore*, *move*, *merge*, *delete*, *reverse share*, *move and merge*, and so on.

Review Phase

After the comparison phase has completed, the results can be reviewed, adjusted, and validated before the session is committed. The review phase is where you analyze your comparison and make any necessary adjustments.

In the Review phase of a VCM session, you can

- ◆ Preview what the target view would look like if the session was committed in its current state. The CPC provides this preview as a **Test Perspective**. The [VCMUtility](#) provides a [CheckoutPreview](#) option that allows you to check out files from the preview.
- ◆ Generate a difference report in both the Cross-Platform Client and [VCMUtility](#) that summarizes all item differences and their current actions.
- ◆ Resolve differences requiring user input, and manually merge file contents and property values when there are conflicts. Actions are considered resolved if no user input is required to continue. However, unresolved actions such as differences with merge conflicts, require user input. The session cannot be committed without user attention. As a best practice, all actions should be reviewed and adjusted if necessary.
- ◆ Choose from multiple possible actions to resolve differences where applicable. For example, for an item in a merge state (modified in both views since the last compare), you could choose to overwrite the target item to match the source item instead of choosing to merge the two.
- ◆ Check out files to working folders, perform a build, and run tests on the simulated merge.
- ◆ Define properties of the change package that will be created when the session is saved or committed such as its name and description.

Save and Restore Phase

Optionally, at any time after the compare phase, you can save the VCM session and restore it later. By default, a VCM session is saved as a change package in the target view identified by the session. A VCM session can also be saved as a local VCM session file or exported as a self-contained VCM exchange file. All three ways of saving a VCM session allow it to be restored later. Restoring a session “resumes” the session at the same point at which it was saved.

- ◆ When a session is saved as a change package, you can copy a URL for it to the clipboard and paste it into an e-mail, for example, so others use the URL to open the change package.
- ◆ A session saved as a VCM Session file ([.vcms](#) file extension) is a shortcut that can only be used to restore the session on the same computer. This is because temporary files are saved on the workstation required to resume the session.
- ◆ A session exported as a VCM Exchange file ([.vcmx](#) file extension) is a self-contained “change package” file that can be copied to another computer, e-mailed, checked into StarTeam or added to a CR as an attachment, and so on. Others can then restore, review, and even commit the session.

Starting with the StarTeam 2009 release, change packages are the preferred method for saving and restoring VCM sessions.

Commit Phase

In the commit phase, View Compare/Merge applies changes to the target view based on the decisions made during the compare and review phases, and records the changes in the StarTeam repository as a change package.

Based on session options, various labels can be automatically created that document what changed in the target view. View labels can be created that provide a snapshot of the target view before and/or after the updates were applied. Revision labels can be created that are attached to target view items affected by the VCM session before and/or after the updates were applied.

VCM can also generate a post-commit update report, which enumerates exactly what was changed by the change session.

Auto-merge

The **View Compare/Merge Wizard** has an available option called **Auto-merge Files**. When this option is selected in the wizard, View Compare/Merge automatically merges files without conflicting differences at the beginning of the View Compare/Merge session.

Auto-merge does a 3-way merge. The auto-merge examines the following in the merged file:

- ◆ What is in all three files.
- ◆ What is in the child that is in neither the ancestor nor the root.
- ◆ What is in the root that is in neither the ancestor nor the child.

That means that it ignores lines that exist in both the child and ancestor, but not the root. It also ignores lines that exist in both the root and ancestor, but not the child. The former would be lines that have been deleted from the revision in the root view, and the latter would be lines that have been deleted from the revision in the child view.

It is possible that the merged file would be identical to the target file, even when the source and target files are different.

For example:

Comparisons are made of an ancestor file `foo.txt` with the target (root view) `foo.txt` and with the source (child view) `foo.txt`. The comparisons show that two lines in question were deleted when the ancestor revision moved to the root (target) revision, but were not deleted from the child (source) revision. Therefore these two lines were in both the ancestor and the child, but not in the root revision. In this case, the two lines in question were deleted from the root view revision. The auto-merge did not restore deleted lines.

You can perform a manual merge when you want to restore deleted lines, and you can perform an overwrite if you want the target (root) to exactly match the source (child). Even if the file is resolved with auto-merge, you can select overwrite or merge again manually.

Most Commonly-used Views

Below are the most common views you will use in view comparisons and merges:

Main View	The main view is where new development eventually appears.
Activity View	The activity view is a child view used for a set of related tasks such as the development of a new feature. This view generally ends when the activity is complete.
Release View	The release view is a child view used to maintain a specific version of an application or module. This view generally ends when the corresponding application or module is no longer supported.

Most Common Merge Tasks

The most common tasks associated with merging views are:

- ◆ Allowing an activity view to catch up with new changes from its parent (Rebase).
- ◆ Delivering a completed activity to its parent (Promote).
- ◆ Duplicating a fix or enhancement to one or more development views (Replicate).

Related Concepts

[View Compare/Merge Session Perspectives](#)

[VCM Merge Types, Rules, and Scenarios](#)

[View Compare/Merge Actions](#)

[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)

[Merging Changes from a Child View to a Parent View \(Promote\)](#)

[Merging Changes from a Parent View to a Child View \(Rebase\)](#)

[Merging Changes Between Any Two Related Views \(Replicate\)](#)

[Using Process Items to Merge Related Files](#)

Related Reference

[Conditions for VCM Action Decisions](#)

[View Compare/Merge Status Icons](#)

Change Packages Overview

StarTeam historically provided many features that supported change management (CM), including built-in workflow, customizable workflow, process links, process tasks, and View Compare/Merge (VCM). Now, StarTeam 2009 adds a comprehensive *change package* object, which allows you to track all changes made in a single commit. As a result of this new feature, VCM now uses change packages instead of VCM process tasks to record its changes.

About Change Packages

Change packages improve StarTeam's ability to manage and track updates. Change packages are an evolution of the View Compare/Merge (VCM) feature first introduced in the StarTeam 2006 release. A change package is an object that contains a set changes applied to a target view. To create a change package, a user first starts a View Compare/Merge session, which acts as a staging area where changes are defined, reviewed, and tested.

A VCM session is visible in the target as a change package when the session is saved or committed. When the VCM session is committed, its changes are atomically applied to the target view, and a change package object records the changes. Subsequently, a change package can be replayed, causing the same changes to be applied to another view.

Change packages represent sets of logically-related software artifact changes. A change package typically contains the changes needed to fix a bug or implement an enhancement, but it can include any set of updates defined by a user as a single logical change. A typical change package will contain mostly new file revisions, but it can also include related non-file changes such as marking a change request as fixed or updating a task as complete. A change package can include updates to any number of items of any type supported by StarTeam.

A change package also includes properties that represent metadata, such as a name, who created the change package, its purpose (description), who last modified the change package and when, and so forth. When a change package is committed, it is stored with the transaction ID with which its changes were applied. The transaction ID allows audit records and command traces associated with the change package to be identified.

When showing changes for a selected item, the Cross-Platform Client will show both legacy and enhanced process links as well as links to relevant change packages. Links to change packages are more granular than links to process tasks were. For example, links are created for update items other than files and folders, and links are created for updates not currently covered by process links such as deletes. The changes defined by a change package are applied as an atomic transaction which means that all of the changes are applied, or none of them are applied.

Because change packages are persistent, you can browse change packages and their details. You can view all change packages applied to a view in chronological order, and for a given change package, you can see both item- and non-item (metadata) change details. You can also query the change history of an item to determine which change packages, if any, were the source of any updates.

Note: Manual links cannot be created to change packages.

The Change Package Object

A change package is a first-class StarTeam object that is persisted in the repository's database. A change package object is used to store information defined in a VCM session when it is saved. In this state, a saved VCM session is presented to users as owned by the target view. When the VCM session is committed, it is still presented as belonging to the target view, but it is presented as a committed change package that can be queried and reused in additional ways.

A VCM session can be created at the client and committed without ever being saved, consequently it is possible to create a change package object in the same transaction that applies its updates. When this occurs, the VCM session is never shown as a saved change package. This is desirable for quick change sessions, for example, a VCM promote of a few items.

- ◆ A change package is owned by a single view, the target view. It is on par with other view-owned objects such as labels and promotion states.
- ◆ A change package cannot be copied, moved, or shared to another view (although it can be replayed as described elsewhere.)
- ◆ Because a change package cannot be shared, the tip revision is always shown in a “current” configuration of the target view, and it cannot be pinned. Consequently, it does not have configuration properties such as Branch On Change or Configuration Time.
- ◆ Change packages do not reside in folders. All change package objects belong to the view object and do not have path-related properties.
- ◆ Change packages are versioned objects, consequently a new revision is created each time an existing object is updated. A change package is updated whenever a VCM session is saved or committed. Essentially, the pre-committed change package becomes the committed change package. A committed change package is read-only and can no longer be changed (except for its revision comment).
- ◆ Change packages behave correctly with respect to “time travel”. That is, when a user sets the target view configuration back in time, and change packages will appear as they did as of the rolled-back time.
- ◆ Change packages can be attached to view labels. This allows, for example, users to attach a change package to a build label to indicate it was included as part of that build. Label attachment also allows two view labels to be compared to see their differences in terms of change packages. A change package also has a Committed in Build property which is set to “Next Build” when the change package is committed. The property is automatically modified to point to the next build label created. This feature allows change packages to identify the first build in which their changes were included.
- ◆ Prior to being committed a change session can be deleted. As with all versioned objects, the delete is “soft” meaning the change package will no longer appear in the tip revision of the target view. However, if the target view is rolled back in time, a previously deleted change package will “reappear” and can be opened in read-only mode. However, there is no way to “undelete” a change package. After a change package is committed, it cannot be directly deleted. Committed change packages are deleted only when the target view is deleted. Database information no longer needed by a deleted change package is removed during the database purge process.
- ◆ Change packages can have queries and filters. This allows change packages to be displayed in the Cross-Platform Client with popular sorting, grouping, and filtering criteria.

In addition to the StarTeam server’s persistence and versioning service, change packages also use the server’s locking service. This means that a change package can be locked exclusively or with a shared lock. An exclusive lock is automatically applied to a saved change package when it is opened for editing. A change package can also be opened in read-only mode, though a non-exclusive lock is not applied for this use. Change packages cannot be “flagged” (bookmarked). Internally, change package objects also use attachments to store options and session state.

Change Package Security

Change packages have similar access rights as other view-level objects such as labels and promotion states. The ability to perform specific updates while committing a change session are derived from the permissions of the target view. For example, the ability to check-in a new revision of a file is derived from that permission in the target view.

An administrator can grant or deny the following view-level rights to any user or group:

- ◆ **Create change package:** Users who do not have this right for a given view can start a change session for that view (in the client application), but they can only save it to an external file format; they cannot save it persistently in the database.
- ◆ **Modify properties:** Users who do not have this right for a given view cannot create a new revision of a change package for that view, for example, by re-saving or committing an already-saved change package.
- ◆ **Delete from view:** Users who do not have this right for a given view cannot delete a saved change package in that view.

- ◆ **See change package and its properties:** Users who do not have this right for a given view cannot access change package objects for that view.
- ◆ **Change access rights:** Users who do not have this right can not modify change package access rights.
- ◆ **See history:** Users who do not have this right cannot see historic revisions of a change package.
- ◆ **Set exclusive locks:** Users who do not have this right cannot acquire and exclusive lock on a change package. this means that they cannot edit a saved change package.
- ◆ **Break exclusive locks:** Users who do not have this right cannot break an exclusive lock currently held by another user.
- ◆ **Label rights:** These rights control the ability to change what labels a change package is attached to. They include: **Attach/Adjust view labels**, **Detach view labels**, **Attach/Adjust revision labels**, and **Detach from revision labels**.

Note: Change package access rights are enforced with the same group/user/container resolution hierarchy as for other security ACLs.

Change Package Replay

You can select a committed change package and apply the same changes to another view. This capability is termed *replay*. A change package selected for replay is termed the *source change package*, and the view that it originally updated (originally the target view) is termed the *source view*. The new view to which the updates are to be replayed is termed the *replay target view*. The replay is identical in function to a VCM merge in which the items updated by the source change package are selected as the source scope and merged to the replay target view.

When a change package is replayed to another view, a new change package is created for the new target view. The new session acts as the staging area for the replayed changes so they can be reviewed, adjusted, and so forth before being committed to the new target.

- ◆ If the replay target view is a parent of the source view, the replay is performed as a VCM promote.
- ◆ If the replay target view is a child of the source view, the replay is performed as a VCM rebase.
- ◆ If the replay target view is not an immediate child or parent of the source view, the replay is performed as a VCM replicate.

VCM Sessions

VCM sessions become change packages when they are persisted at the server. When you initiate a VCM session in the Cross-Platform Client, or with the VCMUtility, it begins as an "in memory" session only. When the VCM session is committed, a change package is created that stores the session's details.

VCM sessions, and consequently change packages, can be created only for updateable view types, which include root (main) views, variant (branching) views, and non-derived (blank) views. A VCM session cannot be created for a read-only reference view.

When you initiate a VCM session, after the compare phase, the session is populated with changes proposed by VCM after comparing items in a source view to items in the target view. The session is given a default change package name and description based on the source scope and merge type with which the session was initiated.

Before a VCM session's updates are committed, it can be saved to the StarTeam repository as a change package. The change package appears as a special type of object owned by the target view. A saved change package can be opened for further editing and saved any number of times. When its changes are committed, a change package becomes a committed change package and can no longer be edited.

Change Packages can be Saved and Restored

When a change package is first created, it only exists as an “application session” in the Cross-Platform Client or VCMUtility. At any time before it is committed, it can be saved, causing its contents to be persisted in the StarTeam repository. Saving a VCM session does not cause it to be “closed” in the Cross-Platform Client. The user can continue working on the VCM session and save it again at any time. Each save creates a new revision of the underlying change package object. The exclusive lock obtained in order to edit a change session is also retained when the session is saved. Only when the VCM session is closed does the user have the choice to release the exclusive lock.

Change Packages can be Opened and Edited

A saved change session appears as owned by the target view to which it applies. The Cross-Platform Client provides a **Change Perspective** view to see saved change packages and open them. A change package must be opened in order to see details about items that have been compared or updated by the session.

By default, when a saved change session is opened, it is locked exclusively at the server. Consequently only one user can actively edit it at a time. If an exclusive lock cannot be obtained, the user can open the change package in read-only mode or attempt to break the exclusive lock.

Note: Breaking an exclusive lock requires the “break lock” privilege.

A user can always open a change package to which he has the read access right in read-only mode. A change package can be opened in read-only mode by multiple users. A change package opened from a rolled-back configuration of the target view can only be opened in read-only mode. Furthermore, the change package revision saved as of the rolled-back configuration is the revision opened. (This is useful, for example, to review information in an older revision of a change package.)

A change package can be opened, edited, and saved any number of times. When a change package is opened, it appears in a Cross-Platform Client window as a resumed VCM session.

When a change session is opened for editing, the only item-level changes allowed are changing VCM action types (for instance, marking a difference as **ignore**) and resolving merge conflicts (for example, manually merging properties or file contents).

The only non-item edits allowed in a change package are:

- ◆ The change package’s Name and Description properties
- ◆ Pre- and post-commit revision and view label options
- ◆ The change package’s Revision Comment, which is used when the VCM session is saved or committed
- ◆ The VCM session’s Working Folder
- ◆ The VCM session’s Default Comment, which is used as the revision comment for any new item revisions created during commit
- ◆ The VCM sessions Responsibility, which is the person responsible for that VCM session

Related Reference

[Change Package Access Rights](#)

Advantages of Change Packages

Historically, View Compare/Merge created a process task when committing a VCM session. Now, VCM no longer creates a process task when it commits a VCM session. Instead, it creates a change package object for tracking the changes in a VCM change session.

The change package is created in the target view and is not shared elsewhere. It is linked to all the items that were affected by the VCM session.

Other advantageous of change packages include:

- ◆ The comprehensive change package object allows you to track all changes made through VCM sessions.
- ◆ Change packages can be saved before being committed so that others can review the proposed changes before they are committed. This can be done by sending a URL to a saved change package in an e-mail, for example.
- ◆ Every committed change package can be “replayed” to other views to which you want to apply the same set of changes.
- ◆ Perspectives in the StarTeam Cross-Platform client window allow you to switch between the **Standard Perspective** and the **Change Perspective** by clicking icons which are right-justified in the toolbar.
- ◆ Using the **Change Perspective**, a manager, for example, can review all the change package objects and the details around those change packages, see what changes were committed to which views, and review changes in uncommitted change packages being proposed for committal.

Note: Once a change package has been committed, it becomes read-only and can no longer be changed.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[Change Packages Overview](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)
[VCM Merge Types, Rules, and Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)

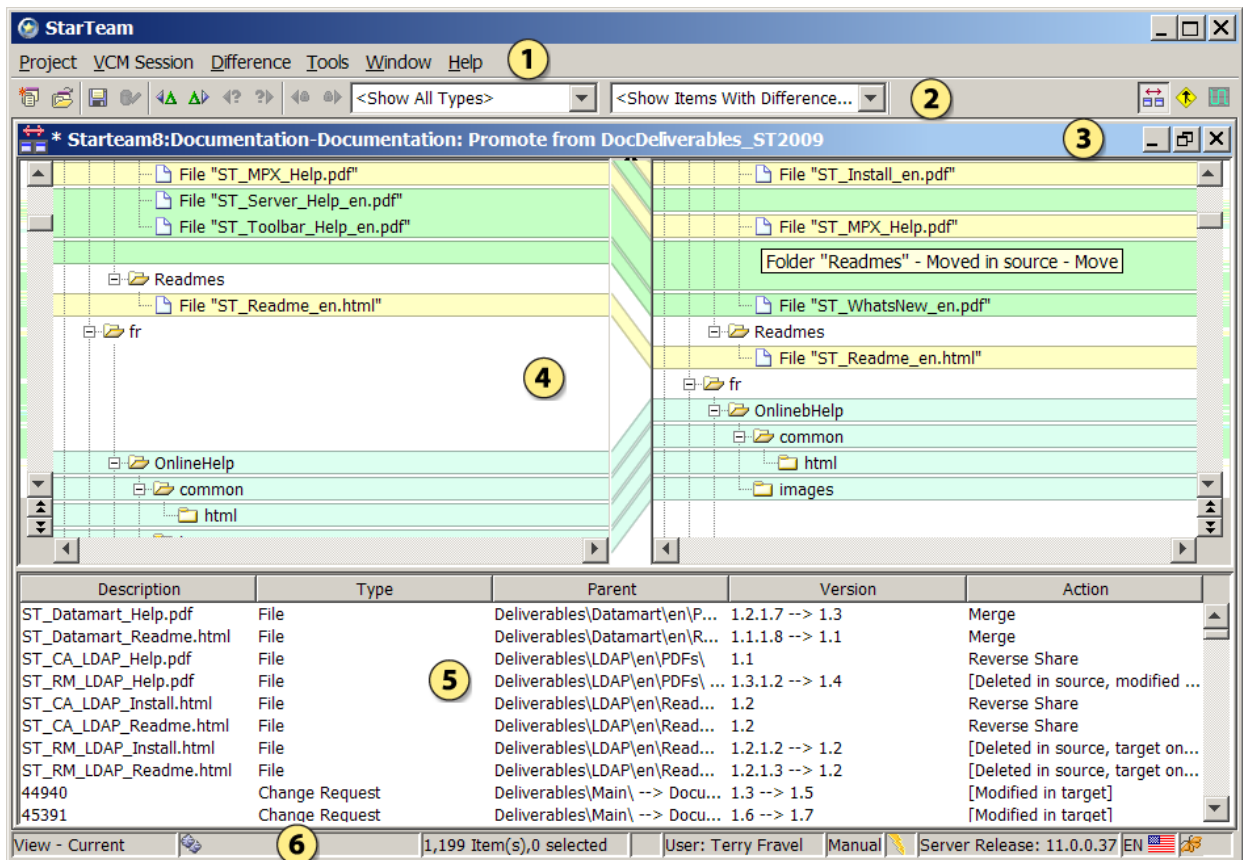
View Compare/Merge Session Perspectives

The View Compare/Merge window provides three perspectives for comparing and merging views, and for resolving item differences.

- ◆ **Compare Perspective**
- ◆ **Merge Perspective**
- ◆ **Test Perspective**

Menu items and toolbar buttons at the top of the View Compare/Merge session enable you to change perspectives. The **Title Bar** of the View Compare/Merge session window displays an icon on the left indicating which perspective is being used.

Compare Perspective



1 VCM Menu

2 VCM Toolbar

3 VCM Title Bar

4 Upper Comparison Pane

5 Lower Detail Pane

6 Status Bar

Use the **Compare Perspective** to review what is being compared and what merge actions are planned. The **Compare Perspective** displays the contents of the source and target views in two folder trees at the top.

Selecting an item in a folder tree also selects it in the lower pane. Each comparison pane contains scroll bars and navigation buttons.

The lower pane in the **Compare Perspective** is a details pane displaying all the items in the two folders. Columns provide the name of the item, item type, parent path, and the version of each item.

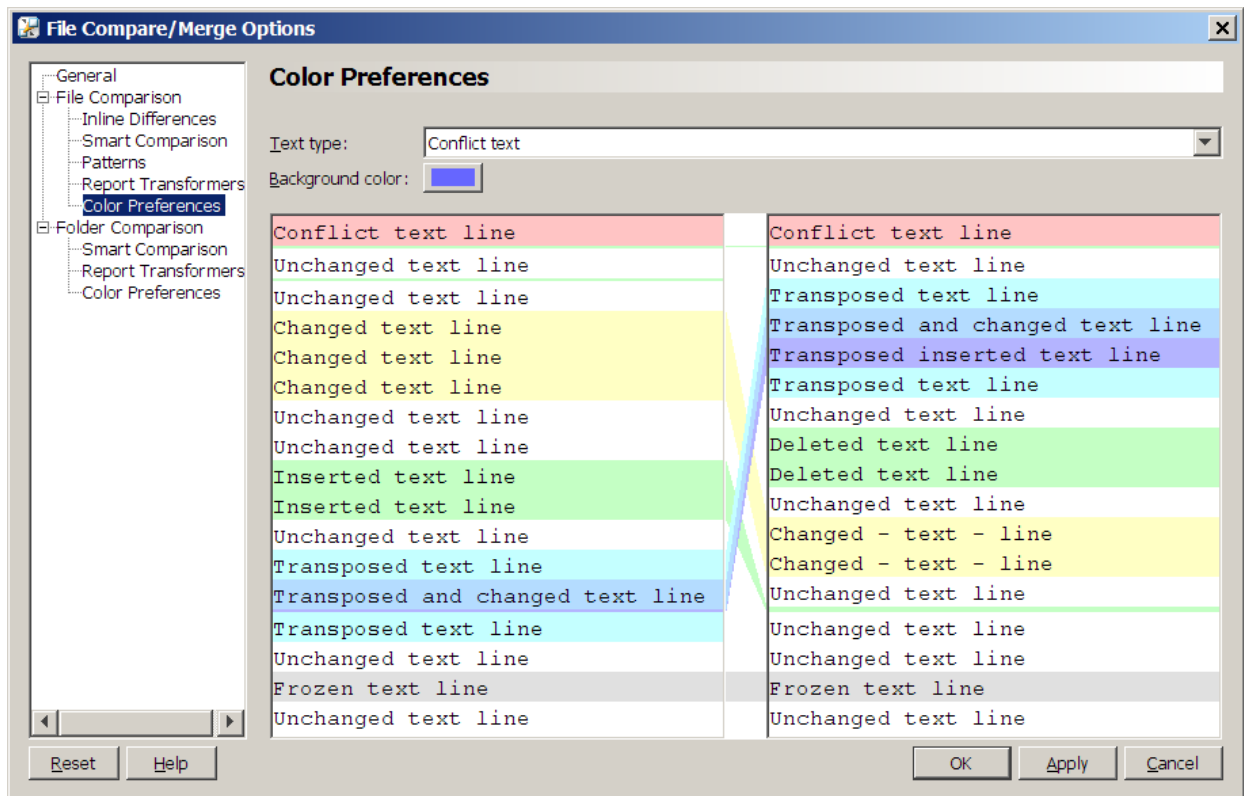
The **Version** column shows both the source and target versions of the item if there is a difference. The **Action** column lists the action to be performed for each item. If the item is not `<unchanged>`, a context menu for the **Action** column enables you to specify a different merge action. The actions are displayed with different formatting indicating the following:

- ◆ An action not in brackets indicates the proposed action View Compare/Merge will take on the item.
- ◆ An action in square brackets indicates the compare phase is ignoring the item, taking no action and displays why in the brackets.
- ◆ An action in angle brackets indicates no action is set for the item because it was not changed or was not compared.

File comparison and folder comparison in View Compare/Merge highlight difference types in color, with each color representing a certain difference type. For example, it highlights inserted or deleted items in green, changed items in yellow, and moved or renamed items in blue. You can modify the default color for differences in the File Compare/Merge Options dialog box, accessed from the File Compare/Merge window under **Tools** ► **Options**.

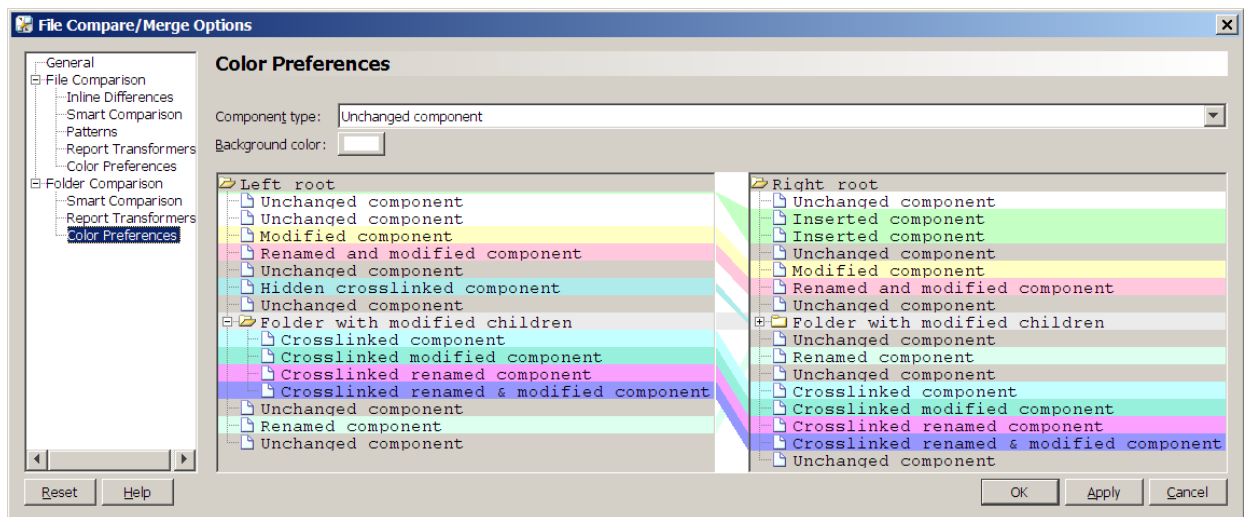
File Comparison Difference Colors

The following image shows the default colors used for highlighting file comparison differences in the **Compare Pane** of View Compare/Merge, and in **File Compare Merge**.

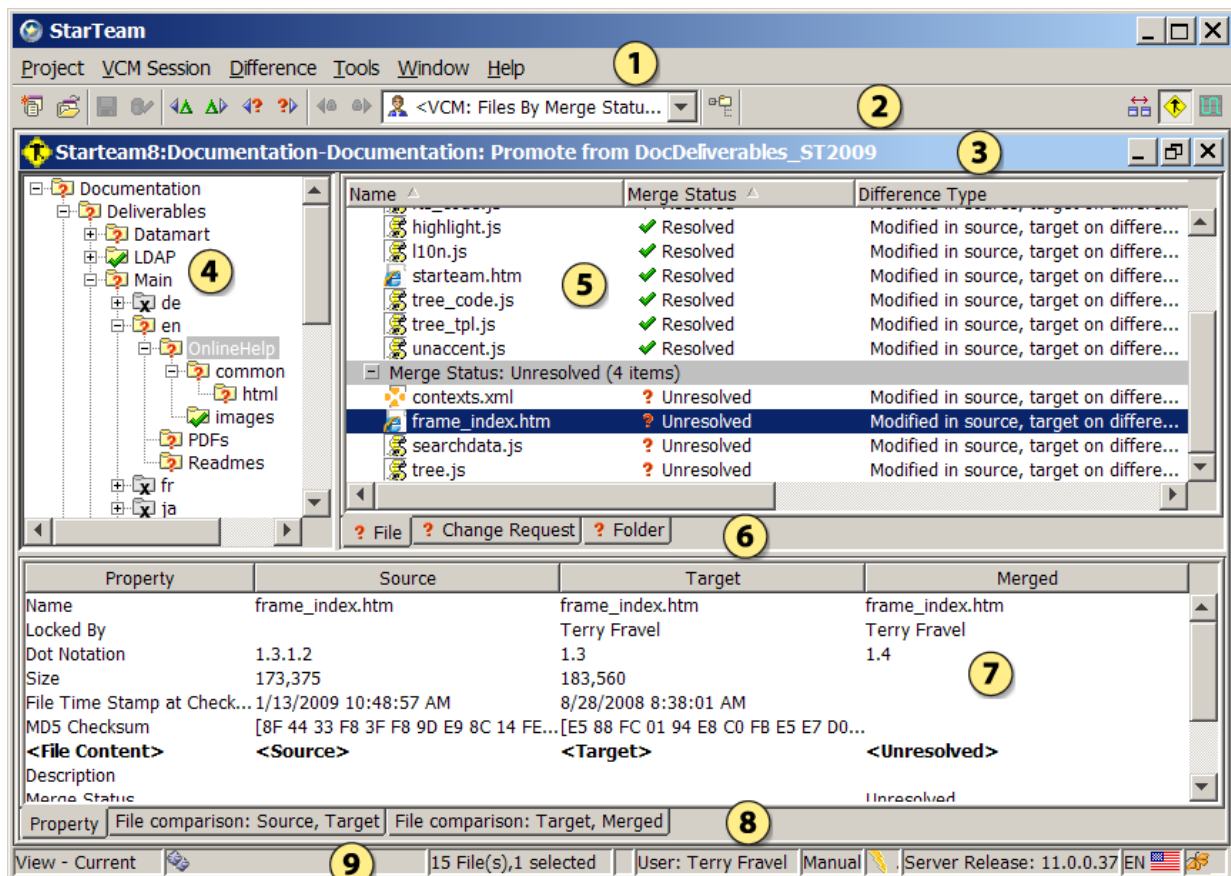


Folder Comparison Difference Colors

The following image shows the default colors used for highlighting folder comparison differences in the **Compare Pane** of View Compare/Merge, and in **File Compare Merge**.



Merge Perspective



- | | | |
|-----------------|------------------|--------------------|
| 1 VCM Menu | 4 Folder Tree | 7 Lower Pane |
| 2 VCM Toolbar | 5 Upper Pane | 8 Information Tabs |
| 3 VCM Title Bar | 6 Component Tabs | 9 Status Bar |

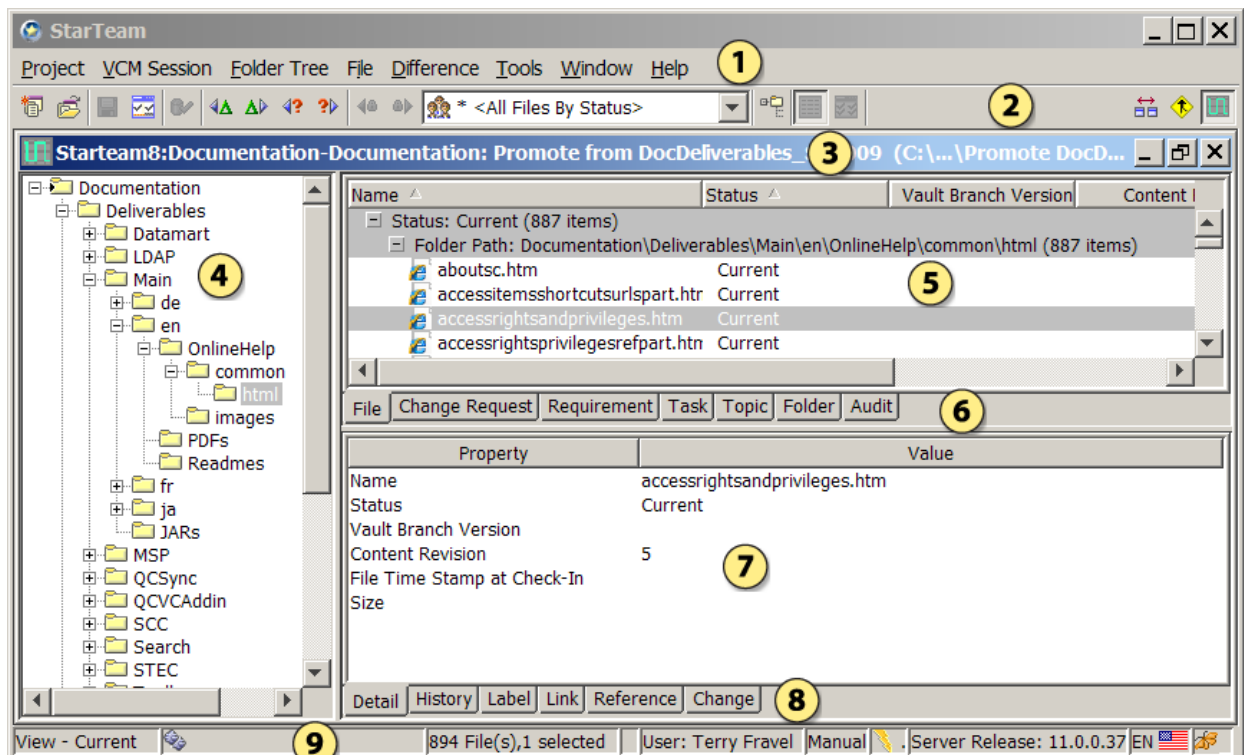
Use the **Merge Perspective** to review and adjust merge results. The **Merge Perspective** displays merge status icons in the folder tree showing which folders have items that need to be merged or resolved. The tabs in the upper

pane are filtered by **Merge Status** or **Merge Action**, and the **Merge Status** column displays the merge status for each item, such as **Unresolved** and **Resolved**. The icons the upper pane display a flyout menu when you hover over them with the mouse.

The lower pane contains a **Property** tab for displaying the properties of the selected item. For unresolved items, the differences are displayed in boldface type. You can double-click an item or use the context menu on a column to resolve the differences by choosing to use the source version, the target version, or merge the differences. If you choose to merge file content differences, the File Compare/Merge window opens in a three-way merge comparison mode allowing you to automatically, or manually perform the merge. You can also merge text properties. For more information, see Overview of Comparing and Merging Files and Folders in the links below.

The lower pane also contains two file comparison tabs, one for comparing a source file with a target file, and one for comparing a target file with the proposed merged results.

Test Perspective



- | | | |
|-----------------|------------------|--------------------|
| 1 VCM Menu | 4 Folder Tree | 7 Lower Pane |
| 2 VCM Toolbar | 5 Upper Pane | 8 Information Tabs |
| 3 VCM Title Bar | 6 Component Tabs | 9 Status Bar |

The test perspective shows you what the target view would look like if the View Compare/Merge session were committed now. You can do many of the same StarTeam operations in the **Test Perspective** as you would do in a normal project view, such as checking out files, checking in files, viewing properties, viewing histories, and comparing changes between revisions.

In the testing phase of a View Compare/Merge session, you can check out files after the compare phase, after auto- and manual-merging has occurred, but before a commit occurs. By default, files checked out from the **Test Perspective** are saved in a temporary View Compare/Merge session working folder in the `[user_home]\Borland\StarTeam\vcRootDir` directory. You can specify a different working folder if you do not want to use the default. You can run your build tools on this folder to test your changes before committing and closing the session.

Once you have determined that the changes from your View Compare/Merge session are correct, you can commit your changes. View Compare/Merge applies all the changes to the StarTeam repository on the server.

Related Concepts

[View Compare/Merge](#)

[Overview of View Compare/Merge \(VCM\)](#)

[View Compare/Merge Status Icons](#)

[View Compare/Merge Actions](#)

[VCM Merge Types, Rules, and Scenarios](#)

[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

[Changing View Compare/Merge Perspectives](#)

Related Reference

[View Compare/Merge Wizard](#)

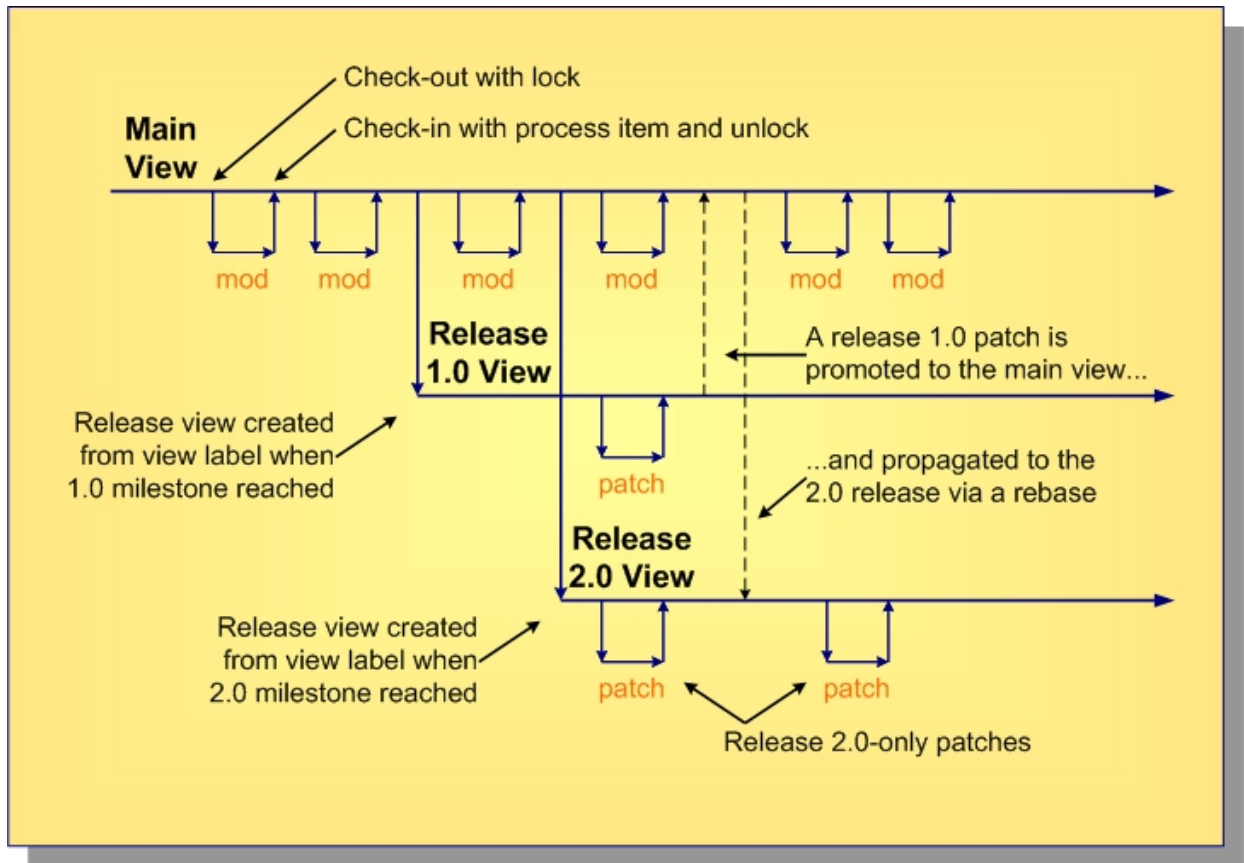
View Merge Type Scenarios

The topics below present examples of some typical full-lifecycle scenarios using VCM for cross-view change management.

VCM Usage with a Simple Project

In this example, you have a small project and a small team, making it is generally safe (for example, not disruptive) to perform new development directly in the main view. The only other views you may need are release views for maintenance purposes. In this scenario, your primary use of VCM is to propagate patches from or to the release views.

Below is an example of using VCM with a simple project.



In this scenario, you perform all new development with direct modifications in the main view. When a release milestone is reached, you create a view label as the reference point for defining a release view. To fix bugs in a given release, you make changes directly in the release view. If you make a patch in one release that must be propagated to the main view and another release, first deliver it to the main view using a VCM promote, then propagate it to the second release with a VCM rebase. (Alternatively, if the patch did not apply to the main view, you could use a VCM replicate to propagate it directly from one release view to another.)

Using VCM with Activity and Sandbox Views

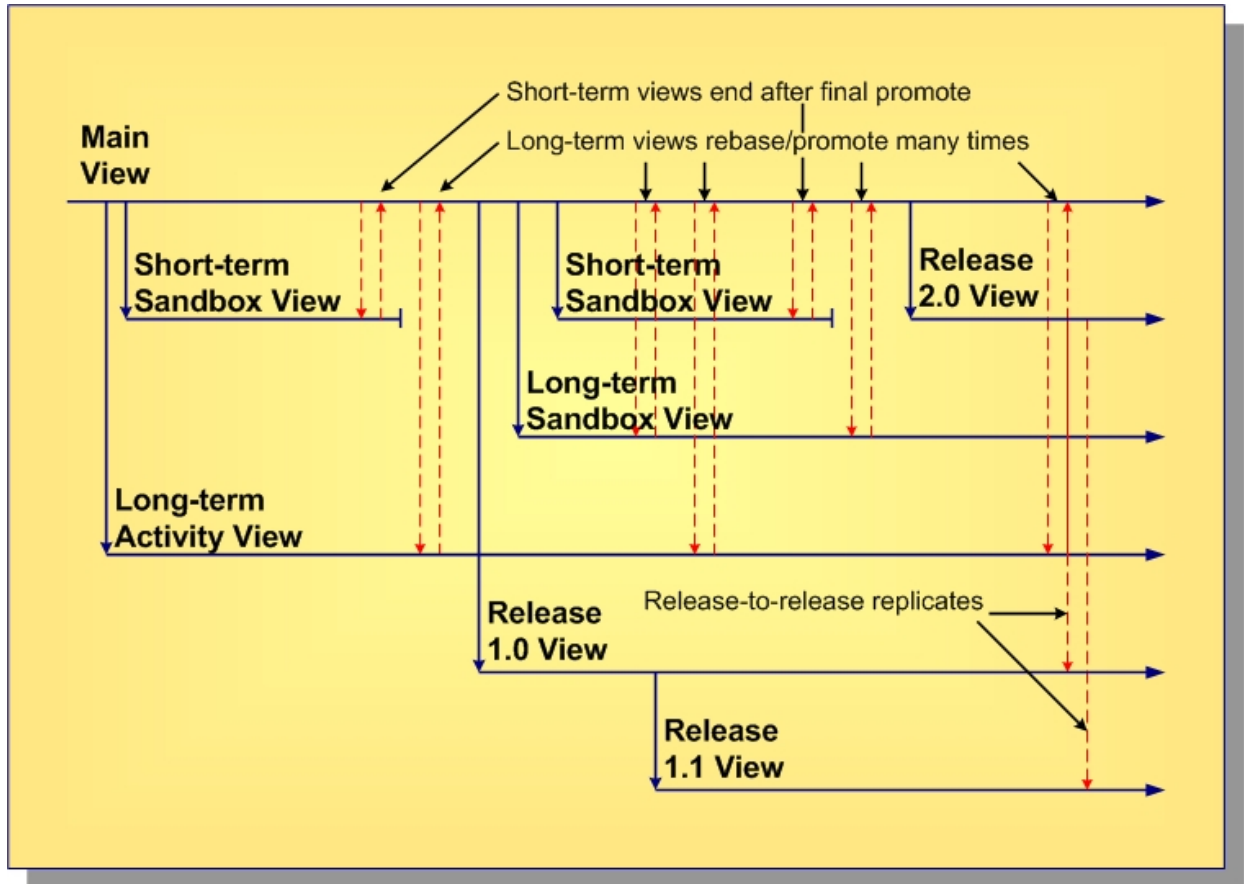
In most circumstances, direct modification to the main view is not advisable due to the potential instability it may cause. Activity and sandbox views provide a place where development can occur without disrupting the stability of the parent view. These view roles also provide a way to discard work when there is a chance that it could be cancelled and never delivered to the main view.

The primary difference between activity and sandbox views is scope:

- ◆ Activity views generally serve a group of developers and a specific purpose such a single, large enhancement.
- ◆ Sandbox views generally serve an individual developer or a very small team and can be used for many iterations of bug fixes and enhancements.

Both activity and sandbox views can have short or long life spans.

When activity and sandbox views are used, all new, fully-verified and approved changes should make their way to the main view. Release views should be created from the main view to provide streams for maintenance work. It even makes sense to sometimes create a release view from another release view, for example when a “dot release” will be shipped then maintained separately from its parent release. An example of recommended VCM practices with activity and sandbox views is shown below.



In this example, two short-term sandbox views are created from the main view. (The “mod” and “patch” cycles shown in the previous example are omitted here to reduce clutter.) After the work is complete in these views, they are promoted to the main view and become inactive (and eventually deleted). A long-term activity view and a long-term sandbox view are also shown. These views “live” beyond a single release. Therefore their work may be promoted to the main view several times.

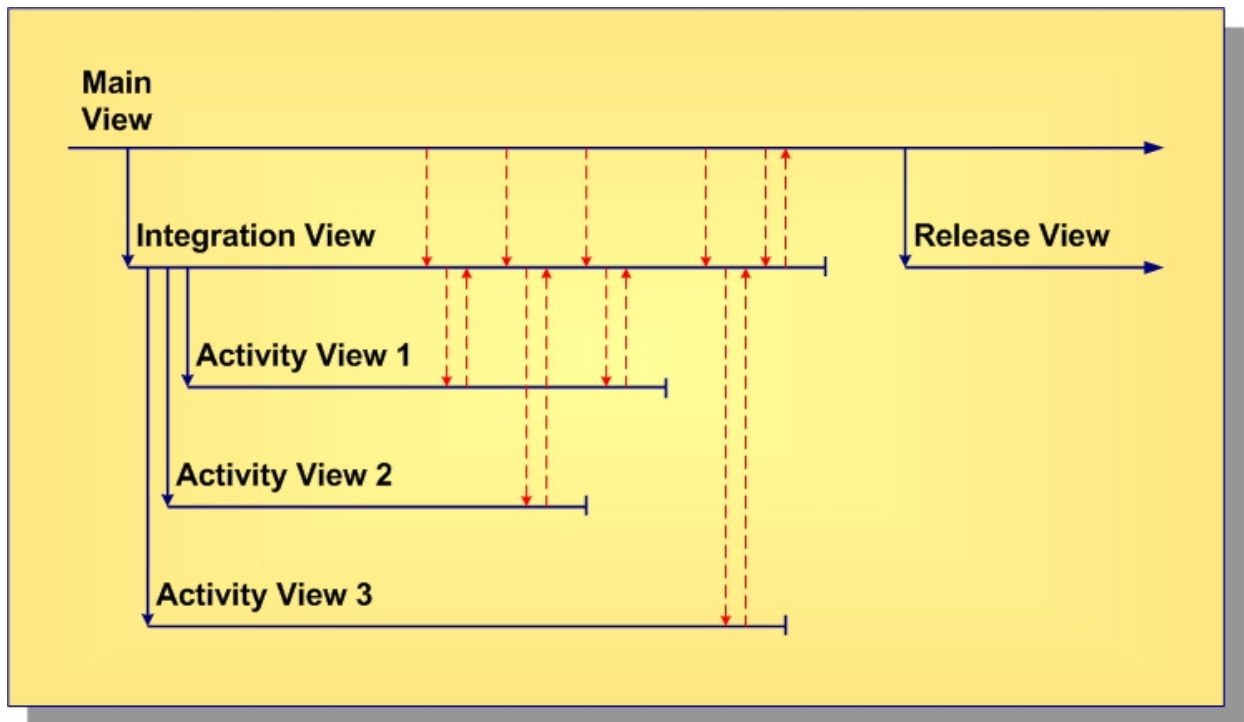
This example demonstrates the recommended best practice of always rebasing a view just prior to promoting it. This causes the child view to merge changes from the parent so they can be integrated and tested.

This example also shows release 1.0 and 2.0 views being created from the main view, but a release 1.1 view is created as a child of the release 1.0 view. Also shown is a patch made in the release 2.0 view and being “back ported” directly to the release 1.0 and 1.1 views using replicate merges.

Using VCM in High Concurrency Projects

Suppose you have a large project involving many sub-teams, each developing a separate component of a final application or suite. You may find it impractical to have each team use its own activity view and then integrate the changes in the main view. The integration and stabilization of large projects can be lengthy, causing long periods of instability in the main view, which is not good.

To address the needs of large development projects, we introduce the notion of an *integration view*. An integration view is a branch-all view that is positioned between the main view and one or more activity views. When the time is right, the “grandchild” activity views are promoted to the integration view, where integration testing occurs. When the integration view passes all applicable tests, it is finally promoted to the main view, delivering its changes. An example using integration views is shown below.



Notice that the integration view is created first as a child of the main view. An activity view for each major component or sub-team is then created as a child of the integration view. The reason for creating the integration view first as a parent of the activity views is so that VCM *promote* merges can be used to propagate changes from the activity views. Promote is the best merge technique for new development because it uses rules that are designed for propagating changes “up” the view hierarchy. For example, new items are reverse-shared instead of shared, causing the parent view to take ownership of the main (1.x) artifact branches. If the integration view was created after the activity views, it could not be their parent, so you would have to propagate changes with VCM *replicate*, which uses *share* as the default action for new items instead of *reverse-share*.

If changes occur in the main view while work is done the activity views, the activity views should periodically be rebased to accept those changes. However, they cannot be rebased from the main view since it is not their parent. Instead, you must rebase the integration view first. Rebasing the integration view should be very easy (no merge conflicts should be found) since no changes should occur directly in the integration view until the integration work is ready to begin.

After an activity view has been rebased, it can be promoted to deliver its changes to the integration view. You could immediately retire an activity view once it has been promoted, choosing to resolve any integration issues and fix bugs in the integration view. However, for complex integrations, you may choose to promote, rebase and adjust, then promote an activity view several times. Eventually, though, you should shift the integration effort to the integration view since it has the big picture. When the integration view passes all tests, promote it the to main view, and create a release view.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[View Compare/Merge Actions](#)

[View Compare/Merge Session Perspectives](#)

[VCM Merge Types, Rules, and Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)

VCM Merge Types, Rules, and Scenarios

This topic presents an overview of the different VCM merge types you can use, explains the rules that apply to them, and gives you example usage scenarios for each one.

Merge Types

There are three types of merge operations possible for a VCM session: *Rebase*, *Promote*, and *Replicate*. Each merge type has specific rules that apply to it.

You can perform successive Rebase, Promote, and Replicate operations between the same two views, even with reversing the direction for Replicate. You can only use one merge type for each VCM session, but you can perform as many VCM sessions as you like in any order.

When to Use Each Merge Type

Each of the VCM merge types is intended to be used in specific scenarios. Which merge type you choose for a particular VCM session depends on what you are trying to merge. Using VCM for the right scenarios is important because VCM uses different rules for handling differences depending on the type of merge. Below is a description of the intended use for each VCM merge type and the rules that apply.

Rebase

Use a Rebase merge to “catch-up” a child view with changes that have occurred in the parent view since the child view was created, or since the last Rebase.

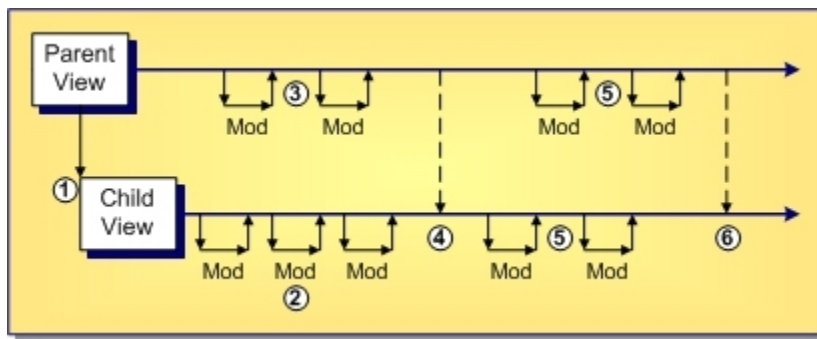
Rules for Rebase

When performing a Rebase merge operation, follow these general rules:

- ◆ The target view must be a branching (variant), immediate child of the source view.
- ◆ The target view items should have branch on change set, and have been created using a pinned (not floating) configuration.
- ◆ The Rebase operation can use any configuration of the source view (tip, label, timestamp, or promotion state), called the *Rebase point*.
- ◆ The Rebase operation uses the tip (updatable) configuration of the target view.
- ◆ Items newly-shared to or re-pinned in the target view use the Rebase/Replicate point. That point is the configuration time of the item in the source view.

Rebase Example Scenario

This example shows a typical scenario where an *activity* (child, branch-all variant) view is created from a specific label, promotion state, or timestamp to support a specific bug fix or enhancement. After changes are made in parallel in the parent and the child, a Rebase is performed to propagate parent changes to the child view. This process is repeated until the child view is no longer needed.



- 1 An "activity" (child, branch-all variant) view is created from a specific label, promotion state, or timestamp to support a specific bug fix or enhancement.
- 2 Changes are made for the activity: modified files branch, new files are added, files are moved or renamed, and so on.
- 3 Changes are made in parallel in the parent view.
- 4 A Rebase is performed to propagate changes in the parent to the child view.
- 5 More changes occur in both views in parallel.
- 6 Another Rebase is performed to propagate new changes.

In this scenario, an activity view is rebased twice to update it with changes that occurred in the parent.

Promote

Use a promote merge to "deliver" changes in a child view to its parent. You could promote once at the end of the child view's lifecycle as a prelude to discontinuing work in the child view. However, you can also promote multiple times to periodically deliver changes from the child view to the parent. In either case, it is typically a good idea to rebase the child view just prior to promoting it to ensure it has "accepted" (been merged with) all recent changes from the parent view.

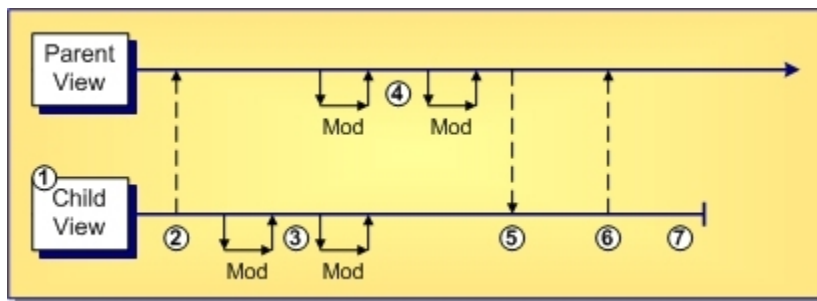
Rules for Promote

When performing a Promote merge type, follow these general rules:

- ◆ The source view must be a branching (variant), immediate child of the target view.
- ◆ The Promote operation uses the tip (updatable) configuration of the target view.
- ◆ The Promote operation can use any configuration of the source view (tip, label, timestamp, or promotion state).
- ◆ New source items are *reverse-shared* to the target view. This means they are moved from the child to the parent and then shared back to the child with **Branch-On-Change** set, and with a pinned configuration. New items not promoted at the latest (tip) revision may be marked as merge conflicts in the child view.

Promote Example Scenario

This example shows a typical scenario where the a child view is created to enable a new branch of development that is simultaneous with continued development on the main branch. During development, the child view is modified, rebased, compiled, tested, then ready for release. At various stages, changes from the child view are promoted to the parent view to release changes, perhaps at key delivery milestones. At the end of the branch's life, a Rebase is performed with the parent view. After the changes in the child from the parent view are compiled, the child view is promoted to the parent again.



- 1 An "activity" view has been created, modified, rebased, compiled, tested, and is ready to be released.
- 2 A Promote is performed to release child changes to the parent view.
- 3 More changes are needed in the activity view.
- 4 Meanwhile, the parent is also modified.
- 5 A Rebase catches up the child with the parent.
- 6 Compile and test are OK, so the child is promoted to the parent again.
- 7 The activity view eventually becomes obsolete after that.

In this scenario, the child view is promoted twice and retired (no longer used) after the second promote. Typically, you'd want to delete a retired view after a period of time.

Replicate

Replicate duplicates changes from one view to another. Use a replicate merge to propagate a change when the target view is not an immediate parent or child of the source view.

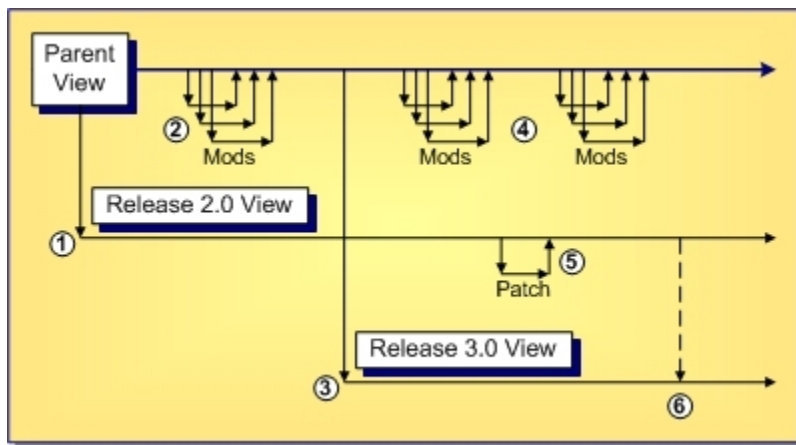
Rules for Replicate

When performing a Replicate merge type, follow these general rules:

- ◆ The source and target views need to belong to the same project. They should have some common lineage, such as parent/child, siblings, uncle/niece, and so on.
- ◆ The Replicate operation can use any configuration of the source view (tip, label, timestamp, or promotion state), called the *Replicate point*.
- ◆ The target view must be an updatable, branching (variant) view, and the tip configuration must be used.
- ◆ Items newly-shared to or re-pinned to in the target view use the Rebase/Replicate point.

Replicate Example Scenario

This example shows a typical scenario where the parent view was branched twice for different product releases, creating 2.0 and 3.0 branches. During product maintenance, changes are made in the 2.0 branch that are applicable to the 3.0 branch, but not to the main branch. A Replicate is performed to duplicate the changes from 2.0 to 3.0.



- 1 A "release" (child, branch-all variant) view is created to support the 2.0 release.
- 2 Changes are made in the main view in preparation for the next release.
- 3 A "release" view is created to support the 3.0 release.
- 4 More changes are made for future releases.
- 5 A bug is fixed in 2.0 that is applicable to 3.0, but not other releases.
- 6 A Replicate is performed to duplicate the patch from the 2.0 view to the 3.0 view.

In this example, a bug fix must be propagated from one release view to another, but the release views are old with respect to the tip revision of the parent view and not applicable to it. Consequently, it makes sense to merge the fix "sideways", directly from the source view to a sibling target view.

View Compare-only Mode

VCM provides a *Compare-only* mode in which you can view the differences between views, but no merge actions or changes are possible.

In Compare-only mode, VCM treats the source view, the target view and the VCM session itself as read-only. Since no changes are made to the source or the target, Compare-only mode supports comparisons that are not otherwise permitted. Specifically, you can compare two different configurations of the same view. For example, you might compare two different labeled configurations, or compare a labeled configuration to the tip.

Related Concepts

- [Overview of View Compare/Merge \(VCM\)](#)
- [View Compare/Merge Actions](#)
- [View Compare/Merge Session Perspectives](#)
- [View Merge Type Scenarios](#)

Related Procedures

- [Comparing and Merging Views](#)
- [Merging Changes from a Child View to a Parent View \(Promote\)](#)
- [Merging Changes from a Parent View to a Child View \(Rebase\)](#)
- [Merging Changes Between Any Two Related Views \(Replicate\)](#)
- [Resolving Merge Conflicts in View Compare/Merge](#)
- [Changing View Compare/Merge Actions](#)

Access Rights Required for StarTeam VCM Users

This document explains the minimum access rights needed by VCM users. It is recommended that project administrators these rights for VCM users since they usually do more with StarTeam than create VCM sessions.

This document assumes that your company uses all item types, which may not be the case. For example, if you do not use or do not have the Requirements component, there is no need for requirement access rights for VCM.

Note: Before you can create access rights for VCM Users, your server administrator needs to set up a VCM Users group. Give your server administrator the list of users to add to this group. Once the VCM Users group is set up on the server, you can open the **Project Access Rights** dialog box ([Project ▶ Access Rights](#)) and set the following access rights for the VCM Users group.

Project Access Rights

The user must be able to see the project that contains the views to be compared and/or merged.

Type of Access Rights	Rights to Grant VCM Users
Generic object rights	See object and its properties

View Access Rights

The user must be able to see the view. Also, since there are options in the VCM that allow the user to create view and/or revision labels, the user should have the rights to create both types of labels.

Type of Access Rights	Rights to Grant VCM Users
Generic item rights	See object and its properties
View specific rights	Create view labels Create revision labels

Change Package Access Rights

Since the VCM session creates a change package object , the user must be given appropriate access rights to change packages.

Type of Access Rights	Rights to Grant VCM Users
Generic item rights	Create change package and modify its properties. Delete change package from view, change access rights, and see history. Set and break exclusive locks.

Label rights

Attach/adjust view labels, detach a view label, attach/adjust revision labels, detach revision labels.

Promotion State Access Rights

No Promotion State access rights are needed.

Child Folders Access rights

The user must be able to see folders and modify their properties because their properties may be merged during the VCM session. For example, a folder's name may have changed.

Type of Access Rights	Rights to Grant VCM Users
Generic item rights	See item and its properties Modify properties Delete from folder Set exclusive locks Break exclusive locks
Label rights	Attach/Adjust view labels Attach/Adjust revision labels
Link rights	See links Create links
Generic item container rights	Share/move folder from its parent folder Modify folder behavior or configuration

Additional Information About Child Folders Access Rights

- ◆ VCM users need the `Delete from folder` access right. VCM or the user may select the **Delete** and **Delete and reverse share** options. Since VCM does not create folders (it only shares and reverse-shares to create new items), the `Create folders` access right is not necessary. However, users of VCM would be very likely to need this right for their everyday work.
- ◆ There are options in the **View Compare/Merge** wizard that require the rights to set and break locks. These options are **Lock target for difference**, **Lock source for difference**, and **Break locks automatically**. Anytime a VCM session results in a reverse share, the user needs the access rights to lock items in both the source and target views. A reverse share can occur during Promote or Rebase operations. A reverse share causes item locking in both views because the item is moved from the source to the target and shared back to the source. So, even if no locking options are chosen when using the **View Compare/Merge** wizard, the user needs the right to lock items in both views any time a reverse share occurs. A reverse share can also result in other changes to the newly shared item in the target. The item's configuration and behavior may have to be reset, and the labels that the item previously had (in the source before the move) are reapplied.
- ◆ Notice that the user needs the right to adjust both revision and view labels. Creating a revision label is not the same as attaching that label to an item. When created, a revision label is attached to nothing. VCM performs attachments as part of the session. While view labels are attached to every item in a view as part of the label's

creation process, users need to be able to adjust those labels. For example, VCM may need to restore the labels to the child folder that the folder had before it was reverse shared.

- ◆ Reverse shares happen during promote sessions and perhaps some replicate sessions. For reverse shares, users need the rights to `Share/Move folder from its parent folder` and `Modify folder behavior or configuration`.
- ◆ The `Modify folder behavior or configuration` access right is also required when the **Fix floating child shares** option is used in the **View Compare/Merge** wizard, and when a re-pin action is selected either by VCM or the user.

File Access Rights

Users need the same access rights for files as for child folders. In addition, they need the rights to check files in and out. Files are checked in as part of the commit portion of a VCM session. Checking in a file requires the user have the ability to see items and properties, modify properties, see item history, and set exclusive locks. Files are checked out by users who use the VCM session **Test Perspective**.

Type of Access Rights	Rights to Grant VCM Users
Generic item rights	See item and its properties Modify properties Delete from folder See history Set exclusive locks Break exclusive locks
Label rights	Attach/Adjust view labels Attach/Adjust revision labels
Link rights	See links Create links
File specific rights	Check in file Check out file
Generic item container rights	Share/move folder from its parent folder Modify folder behavior or configuration

Change Request Access Rights

The access rights for change requests are the same as for child folders, and for the same reasons.

Type of Access Rights	Rights to Grant VCM Users
Generic item rights	See item and its properties Modify properties Delete from folder

	Set exclusive locks Break exclusive locks
Label rights	Attach/Adjust view labels Attach/Adjust revision labels
Link rights	See links Create links
Generic item container rights	Share/move folder from its parent folder Modify folder behavior or configuration

Requirement Access Rights

The access rights for requirements are the same as for child folders, and for the same reasons.

Type of Access Rights	Rights to Grant VCM Users
Generic item rights	See item and its properties Modify properties Delete from folder Set exclusive locks Break exclusive locks
Label rights	Attach/Adjust view labels Attach/Adjust revision labels
Link rights	See links Create links
Generic item container rights	Share/move folder from its parent folder Modify folder behavior or configuration

Task Access Rights

The access rights for tasks are the same as for child folders and they are needed for the same reasons.

Type of Access Rights	Rights to Grant VCM Users
Generic item rights	See item and its properties Modify properties Delete from folder Set exclusive locks Break exclusive locks

Label rights	Attach/Adjust view labels Attach/Adjust revision labels
Link rights	See links Create links
Generic item container rights	Create items Share/move folder from its parent folder Modify folder behavior or configuration

Topic Access Rights

The access rights for topics are the same as for child folders and for the same reasons.

Type of Access Rights	Rights to Grant VCM Users
Generic item rights	See item and its properties Modify properties Delete from folder Set exclusive locks Break exclusive locks
Label rights	Attach/Adjust view labels Attach/Adjust revision labels
Link rights	See links Create links
Generic item container rights	Share/move folder from its parent folder Modify folder behavior or configuration

Note: The details on all access rights for all item types are in topics in the “Reference” section of the Help, under “Administration and Configuration”.

Related Procedures

[Assigning Access Rights to Projects](#)

Related Reference

[Change Package Access Rights](#)

Tips for Successful VCM Sessions

This topic presents some tips for a successful View Compare/Merge session.

Prepare Data

StarTeam allows you to do almost anything you want to your data. However, this can result in conditions which a VCM session cannot make decisions without your input. It is best to get your data into the best possible shape and keep it there:

- ◆ When a branching item (folder, file or CR) is the root branch, or is the first share of a new branch, (1.x), its behavior should be disabled, and its configuration should be floating. Pinning its configuration makes it read-only.
- ◆ When not the first share of its branch, branching items (folders, files and change requests) should be set to branch on change (behavior), and have a pinned configuration.

Some customers make exceptions for this. For example, they might want all change requests visible from all views, regardless of where they were created. So, they share the same change request folder into each child view, but they set the folder and the change requests to NOT branch on change and to be floating. This is not a best practice, but can be used—carefully. Because the change request never branches, when they use a change request from any view as a process item, they are essentially using an out-of-view process item from the main view. That means that they should be using enhanced process links (like all users who use out-of-view process items).

Other customers put all change requests in the main view and always select them as out-of-view process items. They usually make a copy of each change request for each maintenance or release view in which the change request needs to be fixed. This allows each change request for a given problem to take separate workflow steps, or, the same steps but at different times. The copies are also in the main view.

Follow Best Practices

If you follow best practices, the only time you will run into problems is when you share items into a new location. This gives the following results:

- ◆ The shared items have the behavior determined by the view property **Set items shared into view to branch on change** check box, which we recommend to be checked. This property is ignored for items that cannot branch.
- ◆ When dealing with an item that cannot branch (a task, requirement, or topic), pinning its configuration makes it read-only, so you want to use it in only one view at a time. For example, for a task that is created in an activity view, it often makes sense to promote the task along with the changed files. Once the task is promoted, it becomes read-only in the activity view, and you will use it in the parent. This continues up the view hierarchy until the task and its associated changes have been promoted to the root.
- ◆ The shared items will be floating. You will need to set their behaviors to branch on change.

Note: For folders, you select the root folder of the newly shared items (if the share contains folders) and set the configuration for the folder and its subtree of folders.

For the item types in the share, you must select them all from the upper pane (type-by-type), and change their configuration from floating to pinned. Usually customers pin this to the current time, and usually they are primarily concerned with files.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Session Perspectives](#)
[VCM Merge Types, Rules, and Scenarios](#)
[Conditions for VCM Action Decisions](#)
[VCM Table of Action Decisions](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)

Related Reference

[How VCM Handles Target Folders with Floating Shares](#)

View Compare/Merge Action Decisions

This sections contains reference topics explaining View Compare/Merge action decisions.

In This Section

[View Compare/Merge Actions](#)

Presents an overview of how actions View Compare/Merge can take in a VCM session.

[Conditions for VCM Action Decisions](#)

Describes the View Compare/Merge actions.

[VCM Table of Action Decisions](#)

Describes the criteria VCM uses to decide what actions to take for a given merge type.

[Understanding VCM Difference Types](#)

Gives a detailed explanation about some of the difference types you will encounter in a View Compare/Merge session.

[How VCM Handles Target Folders with Floating Shares](#)

Describes how VCM handles floating shares.

[View Compare-only Difference Table](#)

Describes the criteria a View Compare-only session uses to find item differences between views.

View Compare/Merge Actions

When you start a VCM session, it automatically compares the contents of the source and target views using the options you selected in the **View Compare/Merge Wizard**. Wherever possible, VCM suggests a default action to apply to the target view for each item with differences. However, VCM is unable to automatically resolve or merge some types of differences, such as certain conflicts between file contents. If it cannot determine an action, or merge the differences, it gives the items an **Unresolved** status and proposes a **Merge** action. For **Merge** items, you must perform a manual merge or change the action to one that VCM can perform automatically, such as **Ignore**.

You can override the default action proposed by VCM for both **Resolved** and **Unresolved** differences. When VCM cannot suggest a resolution, the action is **Fail** and must be overridden.

VCM can perform many different actions during a VCM session, but it only allows specific actions appropriate for each type of change. For example, if an item is new in the source and does not exist in the target, you can ignore the item, but you cannot delete it from the target. **Delete**, in this case, is not a possible action.

The following table describes each of the actions VCM can perform for a given source/target item difference.

Note: When you create a custom VCM merge type with the **Customize VCM** tool in the **Server Administration**, a `custom.vcm.xml` file is created and each merge action is represented by an enumerated value which is shown in the following table.

Enumerated Value	Action	Description
0	Ignore	Take no action.
1	Re-pin	Changes the revision to which the target is pinned to match the source item.
2	Move	Moves the target item to the equivalent folder as the source item.
4	Merge	Merges the source and target items. The target item is modified to reflect the merged result.
8	Delete	Deletes the target item.
16	Share	Shares the source item to the target view.
32	Reverse Share	Moves the source item to the target view and share it back to the source view.
64	Overwrite	Overwrites the target with the contents of the source.
66	MoveAndOverwrite	Equivalent to a Move followed by an Overwrite .
256	Mark Resolved	Creates a <i>Mark Resolved</i> Change Object. <i>Mark Resolved</i> is now a legal action in many cases where it was not previously permitted. This affects about two dozen difference types.
6	Move and Merge	Equivalent to a Move followed by a Merge .
3	Move and Re-pin	Equivalent to a Move followed by a Re-pin .
40	Delete and Reverse Share	Equivalent to a Delete followed by a Reverse Share .
128	Needs Review	Needs Review is a legal action in all cases. View Compare/Merge cannot determine what action to take and will not commit changes while items are set to Needs Review . You must set Needs Review items to another action, one that can be performed. Changing the default action of any difference type to <i>Needs Review</i> means that human intervention is required before a commit can be made.

Not every action is valid for every item difference. For example, [Delete](#) is not valid when the target item is already deleted.

Note: No **Merge** or **Mark As Resolved** actions are possible for requirements, tasks, and topics, which are times which cannot be branched.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Session Perspectives](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Conditions for VCM Action Decisions

This topic explains the actions VCM can take to resolve differences between views during a VCM session.

VCM looks at the state of each item being compared during the compare phase of a VCM session and determines which set of conditions apply to each item. VCM matches the state of the items being compared to a set of predefined conditions, and chooses an action to take for resolving the conflict.

In more complicated situations, VCM is unable to determine an action to take. In this case, you must specify the action VCM should take to resolve the differences. You may also override the default action chosen by VCM to resolved a conflict.

VCM compares the source and target views to determine which set of conditions apply to the source and target item differences. Below are the conditions for which VCM tests:

Conditions tested	Description
Whether the item already exists in either the source or target views.	These conditions are represented by the Present columns of VCM Decision Table
Whether the item has been deleted from either the source or target views.	These conditions are represented by the Deleted columns of VCM Decision Table.
Whether the item has been modified in either the source or target views.	These conditions are represented by the Changed columns of VCM Decision Table.
Whether the item has been moved in either the source or target views.	These conditions are represented by the Moved columns of VCM Decision Table.
Whether the source or target item is the root branch of its share tree	These conditions are represented by the Root Share columns of VCM Decision Table
Whether the source item is a child share of the target item Whether the target item is a child share of the source item.	These conditions are represented by the Child Share of Twin columns in the VCM Decision Table and refer to an immediate child, not a grandchild, of the matching--that is twin--item in the other view.
Whether the source or target items have a floating configuration	These conditions are represented by the Floating columns of VCM Decision Table.
Whether the target folder has floating shares in the source view.	This condition is represented by the Target folder has floating shares column of VCM Decision Table.
Whether the target item's folder has been deleted.	This condition is represented by the Target Parent Deleted column of VCM Decision Table.
Whether the target item's folder has been ignored by the VCM session.	This condition is represented by the Parent Folder Ignored column of VCM Decision Table.
Whether the target item's folder has failed in the VCM session.	This condition is represented by the Parent Folder Failed column of VCM Decision Table.
Whether the file (if the item in question is a file) is binary.	This condition is represented by the Items are Binary column of VCM Decision Table.
Whether the source and target items have the same user-modifiable properties and, for files, the same data content.	This condition is represented by the Identical Contents column of VCM Decision Table.
Whether the properties that differ between the two items have been excluded.	This condition is represented by the Excluded Properties column of VCM Decision Table.

Related Concepts

[VCM Merge Types, Rules, and Scenarios](#)

[VCM Table of Action Decisions](#)

Related Procedures

[Resolving Merge Conflicts in View Compare/Merge](#)

VCM Table of Action Decisions

When comparing views in a VCM session, VCM compares the state of each pair of items in the source and target views against a table of *difference types* to determine if the items match the criteria in any of the rows. From that, it decides what, if any, are the default and possible actions for resolving each item difference for a given merge type. The resulting list of default and possible actions is what is presented in the UI during the VCM session.

The "Table of VCM Action Decisions" shows the rules VCM uses to decide which actions to take under a given set of conditions for each merge type: **Promote**, **Rebase**, and **Replicate**. The table shows all the possible actions that can be taken in a VCM Session. In the table, the default action is always listed as the first action. Ignoring the item differences is always one of the options.

This table explains how VCM will treat each pair of matching items, one in the source and one in the target. Because the table indicates what actions VCM allows and what action VCM will select by default, it can be considered a set of use cases. The actions allowed by VCM depend on the type of operation (Rebase, Promote, or Replicate), and the following characteristics of the items:

- ◆ Whether an item in the source matches an item in the target (that is, whether the item exists in only one or both views.)
- ◆ Whether the matching items are on the same branch or different branches (based on their dot notations.)
- ◆ Whether the corresponding items are in the same location in the two views (that is, in the same StarTeam folder path; a difference in location indicates that one or both of the matching items has been moved.)
- ◆ Whether additional revisions of the matching items have occurred since the last merge.
- ◆ Whether the target item is a floating child of the source item.

Note: The common ancestor of the matching items or the merge point set for the matching items is used to determine if changes have been made to the source and target items.

Table of VCM Action Decisions

Below is the table VCM uses in for the comparison. An explanation of how to interpret the information in this table follows the table.

Use the following table legend with the table:

Table Legend

Source = Source View

Target = Target View

T = True

F = False

Empty cell = Does not apply or does not matter

Note: Since Rebase and Replicate have very similar options, in most cases the default actions for Rebase and Replicate are the same.

Table of VCM Action Decisions

View Compare/Merge Decision Table																										
Row ID	Description	Merge Types		S		T		P		D		C		M		R		CS		CT		F		T		Possible Actions (First action listed is default)
		Release	Replicate	Promote	Present	Deleted	Changed	Moved	Root Share	Child Share of Twin	Floating	Target Folder has Floating Share	Parent Folder Ignored	Parent Folder Deleted	Different Branches	Identical Content	Excluded Properties									
100	Parent folder failed	X	X	X																						Needs Review, Ignore
110	Parent folder ignored	X	X	X																						Ignore, Needs Review
200	Target folder has floating share in source view	X	X	X																						Needs Review, Ignore (overlaid on other rows)
620	Deleted in target (Promote)		X	T	F	F	T																			Ignore, Needs Review, Reverse Share, Share
600	Deleted in target	X	X	X																						Ignore, Needs Review, Share
520	New in source, shared (Promote)		X	T	F	F																				Needs Review, Ignore, Needs Review, Share
510	New in source (Promote)		X	T	F	F																				Reverse Share, Ignore, Needs Review, Share
500	New in source	X	X	X																						Share, Ignore, Needs Review
800	Deleted in source	X	X	X	F	T	F	F																		Delete, Ignore, Needs Review
900	Deleted in source, target on different branch	X	X	X	F	T	F	F																		Ignore, Delete, Needs Review
1000	Deleted in source, modified in target	X	X	X	F	T	F	F																		Ignore, Delete, Needs Review
700	New in target	X	X	X	F	T	F																			Ignore, Delete, Needs Review
1230	Floating child share, source unbranched (Promote)			X	T	F	F	F	F	F	T															Delete-and-Reverse Share, Ignore, Mark Resolved, Needs Review, Repin
1220	Floating child share (Promote)			X	T	F	F	F	F	F	F															Needs Review, Mark Resolved, Ignore, Repin
1210	Floating child share	X	X	X	T	F	F	F	F	F	F															Repin, Ignore, Mark Resolved, Needs Review
1200	Identical in source and target	X	X	X	T	F	F	F	F	F	F															Ignore, Needs Review
1300	Unchanged since last merge	X	X	X	T	F	F	F	F	F	F															Ignore, Needs Review, Overwrite
1410	Modified in source, target is child share of source (Promote)			X	T	F	F	T																		Delete-and-Reverse Share, Ignore, Mark Resolved, Needs Review, Repin
1405	Modified in source, same content	X	X	X	T	F	F	T																		Mark Resolved, Ignore, Needs Review, Repin
1408	Modified in source, some changed properties excluded	X	X	X	T	F	F	T																		Merge, Ignore, Mark Resolved, Needs Review, Overwrite, Repin
1400	Modified in source	X	X	X	T	F	F	T																		Repin, Ignore, Mark Resolved, Needs Review
1510	Modified in source, target on different branch, same content	X	X	X	T	F	F	T	F	F	F															Mark Resolved, Ignore, Needs Review
1500	Modified in source, target on different branch	X	X	X	T	F	F	T	F	F	F															Merge, Ignore, Mark Resolved, Needs Review, Overwrite
1610	Modified in source, modified in target, same content	X	X	X	T	F	F	T	F	F	F															Mark Resolved, Ignore, Needs Review
1600	Modified in source, modified in target	X	X	X	T	F	F	T	F	F	F															Merge, Ignore, Mark Resolved, Needs Review, Overwrite
1720	Modified in target, same content	X	X	X	T	F	F	T	F	F	F															Ignore, Mark Resolved, Needs Review
1700	Modified in target	X	X	X	T	F	F	T	F	F	F															Ignore, Mark Resolved, Needs Review, Overwrite
1900	Moved in source	X	X	X	T	F	F	F	F	T	F															Move, Ignore, Mark Resolved, Needs Review
2000	Moved in source, target on different branch	X	X	X	T	F	F	F	F	T	F															Ignore, Mark Resolved, Move, Move-and-Overwrite, Overwrite, Needs Review
2050	Moved in source, modified in target	X	X	X	T	F	F	F	T	T	F															Ignore, Mark Resolved, Move, Move-and-Overwrite, Needs Review, Overwrite
2060	Moved in source, branched and modified in target	X	X	X	T	F	F	F	T	T	F															Ignore, Mark Resolved, Move, Move-and-Overwrite, Needs Review, Overwrite
2100	Moved in source, moved in target	X	X	X	T	F	F	F	F	T	T															Ignore, Mark Resolved, Move, Needs Review
2150	Moved in source, moved and modified in target	X	X	X	T	F	F	F	T	T	T															Ignore, Mark Resolved, Move, Move-and-Overwrite, Needs Review, Overwrite
2200	Moved in target	X	X	X	T	F	F	F	F	T	T															Ignore, Mark Resolved, Move, Needs Review
2205	Moved and modified in target	X	X	X	T	F	F	F	T	T	T															Ignore, Mark Resolved, Move, Move-and-Overwrite, Needs Review, Overwrite
2410	Moved and modified in source (Promote)			X	T	F	F	T																		Delete-and-Reverse Share, Ignore, Mark Resolved, Move, Move-and-Repin, Needs Review, Repin
2420	Moved and modified in source, same content	X	X	X	T	F	F	T																		Move, Mark Resolved, Ignore, Move-and-Repin, Needs Review, Repin
2408	Moved and modified in source, some changed properties excluded	X	X	X	T	F	F	T																		Move-and-Merge, Ignore, Mark Resolved, Merge, Move, Move-and-Overwrite, Move-and-Repin, Needs Review, Overwrite, Repin
2400	Moved and modified in source	X	X	X	T	F	F	T																		Move-and-Repin, Ignore, Mark Resolved, Move, Needs Review, Repin
2530	Moved and modified in source, target on different branch, modified in target, same content	X	X	X	T	F	F	T	T	F	F															Mark Resolved, Ignore, Move, Needs Review
2520	Moved and modified in source, target on different branch, modified in target	X	X	X	T	F	F	T	T	T	F															Merge, Ignore, Mark Resolved, Merge, Move, Move-and-Merge, Move-and-Overwrite, Needs Review, Overwrite
2510	Moved and modified in source, target on different branch, same content	X	X	X	T	F	F	T	T	F	F															Mark Resolved, Ignore, Move, Needs Review
2500	Moved and modified in source, target on different branch	X	X	X	T	F	F	T	T	F	F															Merge, Ignore, Mark Resolved, Move, Move-and-Merge, Move-and-Overwrite, Needs Review, Overwrite
2630	Moved and modified in source, moved in target, target on same branch, same content	X	X	X	T	F	F	T																		Move, Ignore, Mark Resolved, Move-and-Repin, Needs Review, Repin
2610	Moved and modified in source, moved in target, target on different branch, same content	X	X	X	T	F	F	T	F	T	T															Mark Resolved, Ignore, Move, Needs Review
2600	Moved and modified in source, moved in target, target on different branch	X	X	X	T	F	F	T	F	T	T															Merge, Ignore, Move, Mark Resolved, Move-and-Merge, Move-and-Overwrite, Needs Review, Overwrite
2615	Moved and modified in source, moved and modified in target, target on different branch, same content	X	X	X	T	F	F	T	T	T	T															Mark Resolved, Ignore, Move, Needs Review
2605	Moved and modified in source, moved and modified in target, target on different branch	X	X	X	T	F	F	T	T	T	T															Merge, Ignore, Mark Resolved, Move, Move-and-Merge, Move-and-Overwrite, Needs Review, Overwrite
2910	Moved and modified in source, moved in target, target is child share of source (Promote)			X	T	F	F	T																		Delete-and-Reverse Share, Ignore, Mark Resolved, Move, Move-and-Repin, Needs Review, Repin
2908	Moved and modified in source, moved in target, some changed properties excluded	X	X	X	T	F	F	T																		Move-and-Merge, Ignore, Mark Resolved, Merge, Move, Move-and-Overwrite, Move-and-Repin, Needs Review, Overwrite, Repin
2900	Moved and modified in source, moved in target	X	X	X	T	F	F	T																		Move-and-Repin, Ignore, Mark Resolved, Move, Needs Review, Repin
2730	Modified in source, moved in target, same content	X	X	X	T	F	F	T																		Mark Resolved, Ignore, Move, Move-and-Repin, Needs Review, Repin
2710	Modified in source, moved in target, target on different branch, same content	X	X	X	T	F	F	T	F	F	T															Mark Resolved, Ignore, Move, Needs Review
2700	Modified in source, moved in target, target on different branch	X	X	X	T	F	F	T	F	F	T															Merge, Ignore, Mark Resolved, Move, Move-and-Merge, Move-and-Overwrite, Needs Review, Overwrite
2715	Modified in source, moved and modified in target, target on different branch, same content	X	X	X	T	F	F	T	T	F	T															Mark Resolved, Ignore, Move, Needs Review
2705	Modified in source, moved and modified in target, target on different branch	X	X	X	T	F	F	T	T	F	T															Merge, Ignore, Mark Resolved, Move, Move-and-Merge, Move-and-Overwrite, Needs Review, Overwrite
2810	Modified in source, moved in target, target is child share of source (Promote)			X	T	F	F	T																		Delete-and-Reverse Share, Ignore, Mark Resolved, Move, Move-and-Repin, Needs Review, Repin
2808	Modified in source, moved in target, some changed properties excluded	X	X	X	T	F	F	T																		Merge, Ignore, Mark Resolved, Move, Move-and-Merge, Move-and-Overwrite, Move-and-Repin, Needs Review, Overwrite, Repin
2800	Modified in source, moved in target	X	X	X	T	F	F	T																		Repin, Ignore, Mark Resolved, Move, Move-and-Repin, Needs Review
9900	Unrecognized	X	X	X																						Needs Review, Ignore

Note: No Merge or Mark As Resolved actions are possible for non-branchable items (requirements, tasks, and topics).

Interpreting the Action Decision Table

The following examples explain how to interpret the rows in this table, reading the columns from left-to-right.

Row 500: New in Source

Conditions and criteria:

- ◆ A file is new in the source view.
- ◆ The merge types for which these conditions and criteria apply are Rebase and Replicate.
- ◆ The file is present in the source, indicated by a **T** (True) under the **Source** header in the **Present** column.
- ◆ The file is not present in the target, indicated by an **F** (False) under the **Target** header in the **Present** column.
- ◆ The file is not deleted in the source, indicated by an **F** under the **Source** header in the **Deleted** column.
- ◆ The remaining columns do not apply to this case and are blank. For example, whether the item is or is not deleted from the target does not matter so the **Deleted Target** column is blank.
- ◆ The possible actions View Compare/Merge can take are to share the new file from the source view to the target view, or ignore the new file (do nothing). The default action will be to **Share**.

Rows 500, 505, 510, and 515 are related but have different possible actions. Sharing is the best option if the source view is the parent view. This is always the case when you are performing a Rebase, and may be the case when you are performing a Replicate. Reverse sharing is the best option if the source view is the child view. This is always the case when you are performing a Promote. Ignoring the item is always an option. In this case, you may not be interested in items that did not exist prior to the creation of a child view or when you last merged these two views.

Row 1200: Identical in Source and Target

Conditions and criteria:

- ◆ A file is identical in the source and target views.
- ◆ The conditions and criteria apply to all merge types: Rebase, Replicate, and Promote.
- ◆ The file is present in both the source and target, indicated by **T** (True) under both the **Source** and **Target** headers in the **Present** column.
- ◆ The file is not deleted in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Deleted** column.
- ◆ The file has not changed in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Changed** column.
- ◆ The file has not moved in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Moved** column.
- ◆ The file is on the same branch in the source and target, indicated by an **F** (False) in the **Different Branches** column.
- ◆ The remaining columns do not apply to this case and are blank.
- ◆ The only possible action View Compare/Merge can take is to **Ignore** the file (do nothing).

Row 1900: Moved in Source

Conditions and criteria:

- ◆ A file has moved in the source view.
- ◆ The conditions and criteria apply to all merge types: Rebase, Replicate, and Promote.
- ◆ The file is present in both the source and target, indicated by **T** (True) under both the **Source** and **Target** headers in the **Present** column.

- ◆ The file is not deleted in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Deleted** column.
- ◆ The file has not changed in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Changed** column.
- ◆ The file has moved in the source, indicated by a **T** (True) under the **Source** header in the **Moved** column, and has not moved in the target, indicated by a **F** (False) under the **Target** header.
- ◆ The file is not on different branches in the source or target, indicated by an **F** (False) in the **Different Branches** column.
- ◆ The remaining columns do not apply to this case and are blank.
- ◆ The possible actions View Compare/Merge can take are to **Move** the file in the target, or **Ignore** the file (do nothing). The default is **Move**.

Rows 1900, 2000, 2100 are related but have different possible outcomes. If the target item is unchanged, the most likely user action would be to move it to the folder in the target view that matches the location of the source item in the source view. If the target item has changed or been moved to yet another location, the most likely user action would be to ignore the target item. The target item may have been moved or modified for reasons totally unrelated to this View Compare/Merge session.

Row 2400: Moved and Modified in Source

Conditions and criteria:

- ◆ A file has moved and has been modified in the source view.
- ◆ The merge types for which these conditions and criteria apply are Rebase and Replicate.
- ◆ The file is present in both the source and target, indicated by **T** (True) under both the **Source** and **Target** headers in the **Present** column.
- ◆ The file is not deleted in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Deleted** column.
- ◆ The file has changed in the source, indicated by a **T** (True) under the **Source** header in the **Changed** column. The target cell is blank because it does not apply in this case.
- ◆ The file has moved in the source, indicated by a **T** (True) under the **Source** header in the **Moved** column, and has not moved in the target (indicated by a **F** (False) under the **Target** header).
- ◆ The file is not on different branches in the source or target, indicated by an **F** (False) in the **Different Branches** column.
- ◆ The remaining columns do not apply to this case and are blank.
- ◆ The possible actions View Compare/Merge can take are to **Move and Re-pin** the file in the target (default), just **Move** the file in the target, just **Re-pin** the file in the target, or **Ignore** the file (do nothing).

Rows 2400 and 2410 are related but have different possible outcomes based on whether or not you are promoting. For example, suppose you are performing a Rebase operation. The item in the source is revision 1.7 and the item has been moved from the Q1 folder to the Q2 folder, while the item in the target is revision 1.5 and remains in the Q1 folder. Because the items are on the same branch, you are more likely to want to do to the target item exactly what has been done to the source item. The default, therefore, is to re-pin the target item to 1.7 and move it to the Q2 folder.

Suppose you are performing a Promote operation. Again, the item in the source is revision 1.7 and the item has been moved from the Q1 folder to the Q2 folder, while the item in the target is revision 1.5 and remains in the Q1 folder. Because this is a Promote operation, you most likely to want a target item in a folder corresponding to the target item's folder and with a matching revision. This is not the same as doing to the target item exactly what has been done to the source item. Because the target view is the parent view and the source view is the child, View

Compare/Merge defaults to a reverse share. The result is that the item appears in the Q2 folder in both views with the reference tree in the appropriate parent/child order. If View Compare/Merge stopped there, the target view would have two copies of the item: the original target item would still be in folder Q1. Therefore, View Compare/Merge also deletes the original target item.

Related Concepts

[VCM Merge Types, Rules, and Scenarios](#)

Related Procedures

[Resolving Merge Conflicts in View Compare/Merge](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Understanding VCM Difference Types

This topic gives a detailed explanation about some of the difference types you will encounter in a View Compare/Merge session.

Below are some of the difference types you may encounter during VCM sessions and their corresponding row numbers in the “View Compare/Merge Action Decision Table”. This section discusses what to do when the difference type results in the default merge action **Fail**. Unless another merge action is selected for items that **Fail**, the VCM session cannot be committed. Often you want to make some data repairs in these cases and then redo the VCM session. Sometimes you will simply ignore the situation temporarily so that you can commit other files. Later, you can perform a repair.

Parent Folder Failed (Row 100)

Defaults to **Fail**.

This difference type is applied dynamically. Whenever an item resides in a folder to which a VCM session has already assigned the merge action **Fail**, then that item also fails, but with difference type **Parent Folder Failed (Row 100)**. If you change the parent folder's merge action, the merge action for the items within that folder will also change.

For example, suppose an item A has the difference type **Parent Folder Failed (Row 100)** because its parent folder B had difference type **Target folder has floating child share in source view (Row 200)**. If you change B's merge action from the default **Fail** to **Ignore**, the parent folder no longer fails. Item A's difference type also changes to **Parent Folder Ignored (Row 110)**. If you want to commit changes to item A, you must fix folder B so that it is not floating in the target view, refresh the view, and redo the VCM session.

Parent Folder Ignored (Row 110)

Defaults to **Ignore** which is also the only available action. This difference type is new with StarTeam 2008 Release 2. It replaces former difference types **New in source, parent folder deleted in target (Row 505)** and **New in source, parent folder deleted in target (Promote) (Row 515)** which also defaulted to **Ignore**.

This difference type is applied dynamically. Whenever an item resides in a folder to which a VCM session has already assigned the merge action **Ignore**, then that item also ignored, but with difference type **Parent Folder Ignored (Row 110)**. This allows you to commit the other resolved items.

However, you can correct the parent folder and also commit the items with this difference type. For example, suppose an item A has the difference type **Parent Folder Ignored (Row 110)** because its parent folder B had difference type **Deleted in target (Promote) (Row 620)** or **Deleted in target (Row 600)**. If the deleted folder belongs back in the target view, you would change B's merge action from the default **Ignore** to **Share** or **Reverse Share**. Because the parent folder issue has been resolved, item A's difference type changes from **Parent Folder Ignored (Row 110)** to a more appropriate difference type. For example, this item may also be **Deleted in target (Row 600)** or it may be **New in Source (Row 500)**. In either of these cases, you may now change item A's merge action to **Share** or **Reverse Share**.

Unlike the **Fail** difference type, **Parent Folder Ignored (Row 110)** usually does not require you to make changes to the repository. Instead, this is used in cases where **Ignore** is the only thing that makes sense, based on a decision you have already made about the parent folder. For example, suppose you have a file that is **New in source (Row 500)**, but the parent folder was **Deleted in target (Row 600)**. The default action for the folder is **Ignore**, which implies that there is no parent folder in the target into which the new file can be shared. That is why the file gets the **Parent Folder Ignored (Row 110)** difference type. If you override the action on the folder to **Share**, the file becomes **New in source - Share**. The **Parent Folder Ignored (Row 110)** difference type no longer applies.

Floating Child Share (Rows 1210, 1220, and 1230)

This section discusses **Floating Share** difference types: **Floating Child Share (Rows 1210)**, **Floating Child Share (Promote) (Rows 1220)**, and **Floating Child Share, Source Unbranched (Promote) (Rows 1230)**. These types occur only when you use the **Fix floating child shares** option in the **VCM** wizard.

The intent of the **Fix floating child shares** option in the **VCM** wizard is to locate items in the target view that:

- ◆ Are on the same branch.
- ◆ Have their parent reference in the source view.
- ◆ Are floating.

The operation type affects which difference type is selected by VCM. If you are not doing a promote operation, the difference type is **Floating Child Share (Row 1210)**. If you are doing a promote operation, the difference type for StarTeam 2008 Release 2 Server and later is **Floating Child Share, Source Unbranched (Promote) (Rows 1230)**. On earlier releases of StarTeam Server, you will see **Floating Child Share (Promote) (Row 1220)** because earlier releases were not able to reverse share items in as many circumstances.

Note: Because of the relationship in the reference tree for the corresponding items in this case, you will never see this difference type when the corresponding items have no common ancestor and were matched only by name.

A floating child item can cause problems. For example, adding an item to a floating child folder in the target can result in that item floating back to the source as a duplicate. Adding a new revision to an item that is floating in the target can result in that revision inadvertently floating back to the source. If the entire view is branch-all floating or branch-none floating, this can happen frequently. That is why StarTeam, depending on how your administrator has set up the server, now either:

- ◆ Restricts floating views permanently.
- ◆ Makes them advanced features to keep them to a minimum.

Floating Child Share (Row 1210)

Defaults to **Repin**. Optionally, you can select **Ignore**.

The best and most common practice in StarTeam is for *parent* > *child* relationships in the share tree to match the *parent* > *child* relationships in the view hierarchy.

This difference type occurs only when you check the **Fix floating child shares** option in the **VCM** wizard. You use this option to stop the creation of unwanted duplicates in the source as an item floats from the target back into the source.

Tip: It is usually better to correct your data before using VCM so that you do not have inappropriate floating shares. Unwanted duplicates can float into more views than the source view.

The intent of the **Fix floating child shares** option in the **View Compare/Merge** wizard is to locate items in the target view that have their parent reference in the source view, and are also floating. Because of this, the rebase operation finds items in the child view that correspond to items in the parent view, and pins them if they are floating. The difference type is **Floating Child Share (Row 1210)**. If you allow VCM to repin in this case, it changes the floating items to the configuration time of this particular VCM session when the session is committed. This is a good idea because you will not have problems with these items in the future.

If you are performing a replicate operation, the references in the reference tree and the views in the view tree may not be pointing in the same direction (from parent to child), but the results will be the same. VCM offers to repin these

items in spite of their reference relationship because the two views in the VCM session are probably not parent and child anyway.

Note: This difference type has limitations: While VCM changes the floating child item from a floating to a pinned configuration, it only pins items in the target view. Other views may have floating references for these items. Duplicates can occur in those other views if you proceed with the commit.

For example, suppose a folder was shared from the parent view to each of two child views, set to branch on change in each child view, but allowed to float in the child views. Because VCM alerted you, you changed the folder in the first child view so that it is now pinned. Nevertheless, if you promote an item from that folder in the now corrected child view to the parent view, the item will float down to the other child view from the parent view if no intervention has taken place in that view.

If the creation of floating views was common practice, it is best to pin items in all the project's child views before a VCM commit operation. It is also a good practice to pin floating items after they are shared to a new location.

Floating Child Share, Source Unbranched (Promote) (Row 1230)

Defaults to **Delete-and-Reverse-Share**. Other merge actions available are **Ignore** and **Repin**. This difference type is new with StarTeam 2008 Release 2.

This difference type occurs only in promote operations when you check the **Fix floating child shares** option in the **View Compare/Merge** wizard. You use this option to stop the creation of unwanted duplicates in the source as an item floats from the target back into the source.

Tip: It is usually better to correct your data before using VCM so that you do not have inappropriate floating shares. Unwanted duplicates can float into more views than the source view.

This difference type allows VCM to do more than **Fail** when a floating share is found during a promote operation. If you perform a **Delete-and-Reverse Share**, the reference tree is corrected. Since both items are on the same branch, re-pinning is also a viable option because it does not cause an unwanted duplicate item to occur.

Target Folder Has Floating Share in Source View (Row 200)

Defaults to **Fail**. Optionally, you can select **Ignore**.

This difference type occurs when the item being processed by VCM exists only in the source folder. Without the folder problem, this item might have resolved to one of several different difference types:

- ◆ Share
- ◆ Reverse Share
- ◆ Move
- ◆ A compound action that includes one of the above actions, for example **Move and Merge** or **Delete-and-Reverse Share**.

This situation produces a **Fail** condition because if the item were moved, shared, or reverse-shared in the target, a duplicate copy of the item would float back into the source folder. You may see **Share**, or another of the possible merge actions, in the **Default Action** column for this item. This is because without the floating folder problem, the item's default merge action would have been **Share**. Because of the floating folder problem, VCM overlays the item's difference type with the **Target Folder Has Floating Share in Source View (Row 200)** difference type. VCM elects to **Fail** rather than cause multiple copies of the item in the source folder.

For example, suppose a folder was shared from the parent view to the child view, set to **Branch on Change** in the child view, but allowed to float in the child view. Further suppose that a new file was added to the folder in the child and you are promoting that item to the parent view. Normally, this file's difference type would have been **New in Source** and VCM's default action would be a **Reverse Share**. However, in this case, performing that action would

cause a duplicate copy of the item in the source folder. That is why VCM uses **Target Folder Has Floating Share In Source View (Row 200)** in this case instead of **New In Source (Promote) (Row 510)**.

You can ignore this situation by changing the action for the item to **Ignore** and then commit the rest of the session. Or you can end the session, go to the folder in the source view, and manually change its configuration from floating to pinned. When you redo the VCM session, the item that had the difference type **Target Folder Has Floating Share in Source View (Row 200)** will have a different difference type, such as **New In Source**. The item's default action will be one of the merge actions listed earlier, and there will be no **Fail** condition.

It might be wise to check the other references for this folder to see if they, too, are floating—even if they are in other views—just to be sure that items will not float into other views that are not involved in this operation.

For more information, see "How VCM Handles Target Folders with Floating Shares".

New In Source (Rows 500 through 520)

This series of difference types all have similar names but very different consequences.

New In Source (Row 500)

Defaults to **Share**. Optionally, you can select **Ignore**.

This item exists in the source but has no match in the target view. If this is a Rebase operation, the operation is going in the same direction as the view tree: from parent to child. In a Replicate operation, that may not be the case. Nevertheless, in either operation, VCM shares this item into the target view at its current location in the source and pins its configuration to the time of the VCM session.

New In Source (Promote) (Row 510)

Defaults to **Reverse Share**. Optionally, you can select **Ignore** or **Share**.

This difference type only occurs in promote operations. The only difference between **New In Source (Row 500)** and **New In Source (Promote) (row 510)** is that the VCM operation is going in the opposite direction as the view tree: from the child to the parent.

To be sure that the reference tree for the item goes in the same direction as the view tree, VCM does a reverse share. Next, the item's configuration is pinned to the time of the VCM operation.

New In Source, Pinned (Promote) (Row 520)

Defaults to **Fail**. Optionally, you can select **Ignore** or **Share**. (Obsolete starting with StarTeam 2008 Release 2.)

New In Source, Pinned (Promote) (Row 530) shows up when the item has a deleted root reference, or when it has been shared from a view that is not involved in the VCM operation. The 1.0 revision of this item is never in the location that you find the current item. In the case of **New In Source, Pinned (Promote) (Row 530)**, the item's configuration is pinned. You can **Ignore** the item or **Share** it.

Workaround: For folders, files, and change requests, which are items that can be branched, you must make sure that the items can branch on change, then forcibly branch the files by doing a forced check-in. Sometimes you have to recreate the folder (and its child folders) so that those folders no longer have deleted root references.

In StarTeam 2008 Release 2 or later, this difference type only applies to items that cannot branch (requirements, tasks and topics) that are not the root share. It happens the same way it did prior to StarTeam 2008 Release 2: you must explicitly share the item from somewhere else, and then try to promote it. VCM cannot reverse share, because VCM cannot force the item to branch.

The workaround is the same as before: review the item to make sure you did not create the source share unintentionally, then override the action from **Fail** to **Share**.

In StarTeam 2008 Release 2, folders, files, and change requests resolve to **New In Source (Promote) (510)** instead of either **New In Source, Shared (Promote) (Row 520)** or **New In Source, Pinned (Promote) (Row 530)**. VCM does the workaround for you.

To give you an example, the following figure shows the references for one of 1 files in a folder shared from a view not involved in the VCM session. The entire folder was shared from one child view to another. The reference tree shown for one of the files is the same for the other file and the folder as well.

This difference type only occurs in Promote operations.

Unchanged

When VCM reports a status **Unchanged**, it does not necessarily mean that the items have **not** been changed since the last VCM session.

For example, suppose you change an item in one location and the changes float to another location. If these two items are compared in a VCM session, VCM reports **Unchanged** because the two items are identical.

For example, suppose changes have floated to both the items being compared. and then the child item changes again and branches, VCM perceives that the pair was the same and that then the child changed. It does not recognize that both have changed since the original share or last VCM session.

- ◆ If changes have floated to both items and one is deleted, VCM reports the deletion but not the changes.
- ◆ If changes have floated to both items and then one is moved, VCM reports the move only.

The floating issue can affect many rows of the View Compare/Merge Action Decision table.

Moved

If the files in one view and the files in another view have a common ancestor but are in different locations, they are reported as moved by VCM because a move appears to have taken place. However, that may not be the case. They may have been shared manually from one view to a different location in another view initially. No move has really occurred. What VCM is actually reporting is that they have different locations.

Suppose that all files were shared to a folder other than the corresponding folder in the other view, and were then moved to the corresponding folder.

- ◆ If you share to a folder other than the same parent folder view location, VCM returns a **Moved** identifier, even though the item has not been moved but is still in its original shared location.
- ◆ If you share to a folder other than the parent location, and then move the item to that parent location, VCM does not register the move. Instead, it lists the item as unchanged.

For example, in **Moved In Source (Row 1900)** promote:

- ◆ `Share2Execution` was shared from the `Execution0` folder in View 1 to the `Moveto` folder in View 2, then moved to the `Execution0` folder in View 2. It returned **Unchanged**.
- ◆ `SharerMoveTo` was shared from the `Execution0` folder in View 1 to the `Execution0` folder in View 2, then moved to the `MoveTo` folder in View 2. It returned **Moved in Source**.
- ◆ `Test3` was shared from the `Execution0` folder in View 1 to the `Moveto` folder in View 2 and not moved. It returned **Moved in Source** and the sub-folder was also listed again as **Unchanged**.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Session Perspectives](#)
[VCM Merge Types, Rules, and Scenarios](#)
[Conditions for VCM Action Decisions](#)
[VCM Table of Action Decisions](#)
[Floating Items](#)
[How VCM Handles Target Folders with Floating Shares](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)

How VCM Handles Target Folders with Floating Shares

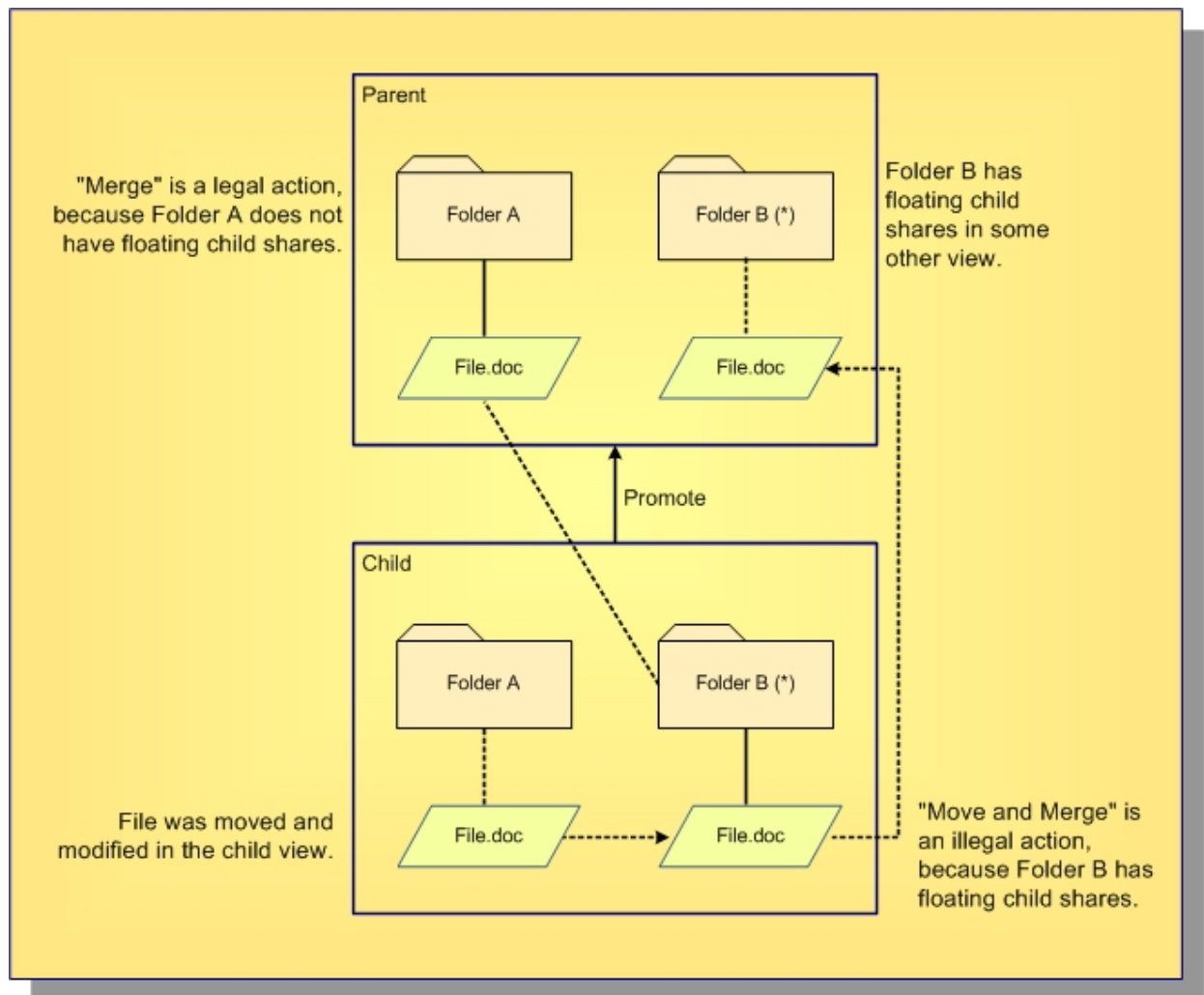
VCM, by its very nature, tends to perform many move, share, and reverse-share operations. If one or more of the target folders has floating child shares elsewhere in the repository, and if VCM performs a share, reverse-share, or move operation, VCM could appear to be creating extra copies of items during the commit. Items could be automatically propagated to other locations throughout the repository, often creating floating child shares unintentionally. In such cases, subsequent operations that trigger automatic propagation can cause unexpected results.

To prevent this from happening, VCM is defensive in the presence of folders that have floating child shares. VCM recognizes situations where an item is being shared, reverse-shared, or moved into a target folder that has active floating child shares, and disallows the operation reporting a *Fail*. To resolve the situation, you can review each failed difference and change the action (for example, to *Ignore*) before committing the session, or correct the floating share problem and redo the VCM session.

Note: When matching an item for which there are multiple shares, VCM prefers matches whose parent/child share relationship is the same as the parent/child relationship between the source and target views.

Example Move Operation in VCM

Move operations present an interesting problem for VCM. Consider the example below.



In this example, an item has been moved and modified in a child view. VCM is then used to promote the changes to the parent view.

Suppose the user were to select Move and Merge as the current action for `File.doc`. In this case, the selected action would move the item into Folder B, which has floating child shares elsewhere in the repository. VCM must therefore prevent the move operation from occurring.

On the other hand, if the user were to set the action to Merge, then the item would remain in its existing folder, and the operation would be permitted. Thus, VCM must recognize the possibility that certain operations may change the target folder, and enable and disable actions appropriately as the target folder changes.

Target Folder Has Floating Shares

The VCM engine has a difference type called “Target folder has floating shares” which is unlike the other difference types in that it is never selected from the VCM Action Decision Table based on the `MatchState` that describes the relationship between a source item and target item. Instead, it is overlaid on top of an existing `DifferenceType` in the context of a given `ItemDifference`, based on an analysis of the target folder that would result from the currently selected action.

Consider again the example above. Use the VCM Action Decision Table rows as the reference, where row [2] is “Target folder has floating shares”, and row [30] is “Moved and modified in source, target on different branch”. In this case, the difference type selected from the decision table for `File.doc` is:

```
Row [30]: Moved and modified in source, target on different branch

Action: Merge
Legal Actions: Merge; Ignore; Move; Mark Resolved; Move and Merge; Overwrite
```

When a client application requests the difference type for this `ItemDifference`, the VCM engine looks at the currently selected action, and determines whether or not a move, share or reverse share will be performed. At this point, the current action is *Merge*, and therefore nothing special happens.

Now, suppose the user sets the action to *Move and Merge*. In this case, when the client application requests the difference type for this `ItemDifference`, the VCM engine notes the move operation and determines whether or not the new target folder has any floating child shares. If it does, then the engine does not return the actual difference type, but instead returns “Target folder has floating shares”.

Furthermore, the VCM engine presents a different current action, and a different set of legal actions. These are determined dynamically, by removing all of the problematic actions and replacing them with Fail. Thus, by overlaying row [2] over row [30], the engine exposes the difference type to client applications as if it were the following:

```
Row [2]: Target folder has floating child shares

Action: Fail
Legal Actions: Fail; Merge; Ignore; Mark Resolved; Overwrite
```

Suppose now that the user changes the current action back to *Merge*. At this point, the engine no longer needs to overlay row [2] over row [30]; it exposes the difference type, current action and set of legal actions in their original, unaltered forms.

Frequency of Checking for Floating Shares

You should note that there is no way to prevent you from adding new floating shares. Thus, no matter when VCM checks, it is always possible that a new floating share is added immediately afterwards, and is missed.

Checking often for a floating child share would decrease the likelihood that we would miss one. However, since this test is lengthy and resource intensive, this would result in unacceptable performance.

Currently, VCM only checks for floating child shares in the source view since the source view is already open, and since checking the source view can be done rather easily and quickly.

To improve performance, VCM checks each target folder at most once per session. However, for sessions that are saved and then later restored, VCM checks again when the session is restored.

Related Concepts

[VCM Merge Types, Rules, and Scenarios](#)

[Understanding VCM Difference Types](#)

Related Procedures

[Changing View Compare/Merge Actions](#)

Related Reference

[Conditions for VCM Action Decisions](#)

View Compare-only Difference Table

When you only want to view the differences between two views, you can run a Compare-only VCM session. In this case, VCM does not determine any default or possible actions to take, since no actions are requested. The Compare-only VCM session simply shows you the differences between the items in the source and target views.

Table of View Compare-only Differences

Below is the table of the criteria that determine the differences between items in a View Compare-only session. An explanation of how to interpret the table follows the table.

Table Legend
Source = Source View
Target = Target View
T = True
F = False
Empty cell = Does not apply or does not matter

Table of differences criteria for a View Compare-only session

View Compare/Merge Decision Table																
Row ID	Description	Present		Deleted		Changed		Moved		Child Share of Twin		Floating		Target Parent Deleted	Identical Content Different Branches	Identical Content
		S	T	S	T	S	T	S	T	S	T	S	T			
600	Deleted in target	T	F	F	T											
500	New in source	T	F	F												
800	Deleted in source	F	T	T	F		F								F	
900	Deleted in source, target on different branch	F	T	T	F		F								T	
1000	Deleted in source, modified in target	F	T	T	F		T									
700	New in target	F	T		F											
1210	Floating child share	T	T	F	F	F	F	F	F		T		T		F	
1200	Identical in source and target	T	T	F	F	F	F	F	F						F	
1300	Unchanged since last merge	T	T	F	F	F	F	F	F						T	
1400	Modified in source	T	T	F	F	T	F	F	F						F	
1510	Modified in source, target on different branch; same content	T	T	F	F	T	F	F	F						T	T
1500	Modified in source, target on different branch	T	T	F	F	T	F	F	F						T	
1610	Modified in source, modified in target; same content	T	T	F	F	T	T	F	F						T	T
1600	Modified in source, modified in target	T	T	F	F	T	T	F	F						T	
1720	Modified in target, same content	T	T	F	F	F	T	F	F							T
1700	Modified in target	T	T	F	F	F	T	F	F							
1900	Moved in source	T	T	F	F	F	F	T	F						F	
2000	Moved in source, target on different branch	T	T	F	F	F		T	F						T	
2050	Moved in source, modified in target	T	T	F	F	F	T	T	F						F	
2100	Moved in source, moved in target	T	T	F	F	F	F	T	T							
2150	Moved in source, moved and modified in target	T	T	F	F	F	T	T	T							
2200	Moved in target	T	T	F	F	F	F	F	T							
2205	Moved and modified in target	T	T	F	F	F	T	F	T							
2420	Moved and modified in source, same content	T	T	F	F	T		T	F						F	T
2400	Moved and modified in source	T	T	F	F	T		T	F						F	
2510	Moved and modified in source, target on different branch, same content	T	T	F	F	T		T	F						T	T
2500	Moved and modified in source, target on different branch	T	T	F	F	T		T	F						T	
2630	Moved and modified in source, moved in target, target on same branch, same content	T	T	F	F	T		T	T						F	T
2610	Moved and modified in source, moved in target, target on different branch, same content	T	T	F	F	T	F	T	T						T	T
2600	Moved and modified in source, moved in target, target on different branch	T	T	F	F	T	F	T	T						T	
2615	Moved and modified in source, moved and modified in target, target on different branch, same content	T	T	F	F	T	T	T	T						T	T
2605	Moved and modified in source, moved and modified in target, target on different branch	T	T	F	F	T	T	T	T						T	
2730	Modified in source, moved in target, same content	T	T	F	F	T		F	T						F	T
2710	Modified in source, moved in target, target on different branch, same content	T	T	F	F	T	F	F	T						T	T
2700	Modified in source, moved in target, target on different branch	T	T	F	F	T	F	F	T						T	
2715	Modified in source, moved and modified in target, target on different branch,	T	T	F	F	T	T	F	T						T	T
2705	Modified in source, moved and modified in target, target on different branch	T	T	F	F	T	T	F	T						T	
2800	Moved in source, moved in target	T	T	F	F	T	F	F	T						F	
9900	Unrecognized	Matches all conditions not covered above														

Interpreting the Table

The following examples explain how to interpret the rows in this table, reading the columns from left-to-right.

Row 500: New in Source

Conditions and criteria:

- ◆ A file is new in the source view.
- ◆ The file is present in the source, indicated by a **T** (True) under the **Source** header in the **Present** column.

- ◆ The file is not present in the target, indicated by an **F** (False) under the **Target** header in the **Present** column.
- ◆ The file is not deleted in the source, indicated by an **F** under the **Source** header in the **Deleted** column.
- ◆ The remaining columns do not apply to this case and are blank. For example, whether the item is or is not deleted from the target does not matter so the **Deleted Target** column is blank.

Row 1200: Identical in Source and Target

Conditions and criteria:

- ◆ A file is identical in the source and target views.
- ◆ The file is present in both the source and target, indicated by **T** (True) under both the **Source** and **Target** headers in the **Present** column.
- ◆ The file is not deleted in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Deleted** column.
- ◆ The file has not changed in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Changed** column.
- ◆ The file has not moved in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Moved** column.
- ◆ The file is on the same branch in the source and target, indicated by an **F** (False) in the **Different Branches** column.
- ◆ The remaining columns do not apply to this case and are blank.

Row 1900: Moved in Source

Conditions and criteria:

- ◆ A file has moved in the source view.
- ◆ The file is present in both the source and target, indicated by **T** (True) under both the **Source** and **Target** headers in the **Present** column.
- ◆ The file is not deleted in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Deleted** column.
- ◆ The file has not changed in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Changed** column.
- ◆ The file has moved in the source, indicated by a **T** (True) under the **Source** header in the **Moved** column, and has not moved in the target, indicated by a **F** (False) under the **Target** header.
- ◆ The file is not on different branches in the source or target, indicated by an **F** (False) in the **Different Branches** column.
- ◆ The remaining columns do not apply to this case and are blank.

Row 2400: Moved and Modified in Source

Conditions and criteria:

- ◆ A file has moved and has been modified in the source view.
- ◆ The file is present in both the source and target, indicated by **T** (True) under both the **Source** and **Target** headers in the **Present** column.

- ◆ The file is not deleted in the source or target, indicated by an **F** (False) under both the **Source** and **Target** headers in the **Deleted** column.
- ◆ The file has changed in the source, indicated by a **T** (True) under the **Source** header in the **Changed** column. The target cell is blank because it does not apply in this case.
- ◆ The file has moved in the source, indicated by a **T** (True) under the **Source** header in the **Moved** column, and has not moved in the target (indicated by a **F** (False) under the **Target** header.
- ◆ The file is not on different branches in the source or target, indicated by an **F** (False) in the **Different Branches** column.
- ◆ The remaining columns do not apply to this case and are blank.

Related Concepts

[VCM Merge Types, Rules, and Scenarios](#)

Related Procedures

[Changing View Compare/Merge Actions](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Atomic Check-ins

All check-ins in StarTeam are atomic. Whenever more than one file is checked in as the result of a single transaction all of the files, and their associated process items, are updated in a single action. If for some reason, the check-in fails, none of the files are checked in, and the status of the associated process items is not updated.

For example, suppose User A selects to check in all modified files in a StarTeam folder, but one of the selected files is locked by User B. Because of the locked file, none of the files are checked in (and none of the process items are updated as fixed) and User A is notified that none of the files were checked in because one of the files was locked by User B.

Quick Access to Projects and Items

StarTeam can open URL shortcut links to projects, views, folders, and items (files, change requests, requirements, tasks, and topics). By creating shortcuts, you can easily and quickly access specific items in a project.

Desktop Shortcuts

If you will be accessing a project view frequently, you may want to save the view as a shortcut on your desktop. Double-clicking the shortcut both starts the application and opens the view associated with the shortcut. You can also create desktop shortcuts to items that you are tracking. Opening the shortcut starts the application, opens the project view in the configuration it had when the shortcut was created, and displays the item's Properties dialog.

URL Shortcuts

You can create URLs and HTML representations for items and copy them to the Windows Clipboard. Depending on the application, a paste operation transfers either the URL or HTML data to the application. For example, you can copy the names of a list of files to a Microsoft Excel spreadsheet using the HTML representation. You can email the URLs for a list of files to a coworker to use in StarTeam, as long as your email application does not convert the paste operation to HTML. Not all applications support pasting the HTML representation, although Word, Excel, and Outlook do support HTML data.

Like other URLs, StarTeam URLs include the name of the server for the connection. In some organizations, StarTeam servers may be reached from both the Internet and the corporate intranet. In such cases, a server may have two different IP addresses. If you configure the server list to reference a server by its IP address, rather than its DNS name, then any URLs generated by the client will work only from the network on which that IP address exists.

The type of URL that your StarTeam client creates for an item is set, per item, on their respective tabs in the Personal Options dialog. There is one exception to this. Because there is no way to set the URL type for a folder, folders always use the ID-based URL type. Also from the Personal Options dialog, you specify the templates used to create an item's HTML representation. For projects, views, and folders there is no HTML representation.

- ◆ If the URL is a reference to a project, the default view of the project opens.
- ◆ If the URL is a reference to a view or folder, then the view or folder opens.
- ◆ If the URL is a reference to an item, the item's view opens, the item's parent folder is selected in the folder tree, the item type is selected, and the item itself is selected in the item list or tree on the upper pane.

The URL can be ID-based or name-based. ID-based is the initial default for each item type. The sample URLs below show the basic differences between ID-based and named-based URLs. They both represent a file in that root view of a project. The ID-based URL is the first of the two.

```
starteam://hostname:49201/12;ns=Project;scheme=id/ 154;ns=View;scheme=id/  
869958;ns=File;scheme=id;scope=full; starteam://hostname:49201/myproject/myview/path to  
myfile;scope full;
```

The advantage of using the ID-based URL is that an item can be moved to a different folder (or a file's name can be changed) and the item can still be located. The advantage of a name-based URL is that the URL can resolve to different StarTeam objects at different points in time. For example, if a file is deleted and then added again (with a new ID), it can still be located.

Related Concepts

[Personal Options](#)

[Toolbar Utility](#)

Related Procedures

[Creating Shortcuts](#)

Procedures

This section contains all the tasks associated with administering and using StarTeam.

In This Section

[Logging On and Off](#)

This section contains tasks related to logging on and off the server.

[Configuring Your Client](#)

This section contains procedures related to configuring the client application.

[Using the Toolbar Utility](#)

This section contains procedures related to using the Toolbar Utility.

[Managing Projects](#)

This section contains procedures related to managing projects.

[Managing Views](#)

This section contains tasks related to managing views.

[Branching Operations](#)

This section contains tasks related to branching.

[Comparing and Merging Views](#)

This section contains procedures related to comparing and merging views.

[Comparing and Merging Files and Folders](#)

This section contains tasks related to comparing and merging files and folders.

[Working with Folders and Items](#)

This section contains procedures related to working with folders and items.

[Managing Files](#)

This section contains tasks related to managing files.

[Working with Change Requests](#)

This section contains tasks related to working with change requests.

[Using Requirements](#)

This section contains tasks related to using requirements.

[Using Tasks](#)

This section contains procedures related to using tasks.

[Using Topics](#)

This section contains tasks related to using topics.

[Viewing the Audit Log](#)

This section contains tasks related to viewing the Audit Log.

[Managing Labels and Promotion States](#)

This section contains tasks related to managing labels and promotion states.

[Linking and Unlinking Items](#)

This section contains tasks related to linking and unlinking items.

[Querying Data](#)

This section contains tasks related to querying data.

[Filtering Data](#)

This section contains tasks related to filtering data.

[Specifying Process Rules and Process Items](#)

This section contains tasks related to process rules and process items.

[Accessing Projects and Items with Shortcuts and URLs](#)

This section contains tasks related to using shortcuts and URLs to access projects and items.

[Working with Revisions](#)

This section contains tasks related to working with revisions.

[Working with Properties](#)

This section contains procedures related to working with properties.

[Creating Reports and Exporting Data](#)

This section contains tasks related to creating reports and exporting data.

[Customizing Reports](#)

This section contains procedural tasks related to customizing reports.

[Creating Charts](#)

This section contains tasks related to creating charts.

[Accessing Projects and Items with Shortcuts and URLs](#)

This section contains tasks related to using shortcuts and URLs to access projects and items.

Logging On and Off

This section contains tasks related to logging on and off of a StarTeam server.

In This Section

[Logging Off](#)

Describes how to log off a StarTeam server.

[Logging On to Servers with the Toolbar Utility](#)

Describes how to log on to a server.

[Logging On to the Server and Starting a Project](#)

Describes how to log onto the server and open a project.

Logging Off

This procedure shows you how to log off a StarTeam server.

To log off a StarTeam server

- 1 Switch the StarTeam client window to a project view on the server on which you want to log off.
- 2 On the StarTeam main menu, choose **File** ► **Log Off**.
A confirmation dialog box appears asking if you want to completely log off this server.
- 3 Click **Yes** to continue.
StarTeam closes all the views on this server that you have logged into during this StarTeam session.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Server Configurations](#)

[Starting Applications with the Toolbar Utility](#)

[Viewing Connection Properties](#)

Logging On to Servers with the Toolbar Utility

This procedure describes the steps for logging onto a server in StarTeam.

Log on to a server

- 1 Choose **Project** ► **Open** to launch the **Open Project Wizard**.
- 2 Select the server in the **Server/Project Tree** and click the **Log On As** button to open the **Log On To** dialog box:

Note: If the server you want is not in the list, click **Add Server** to add a new server configuration.

- 3 Type your **User Name** and **Password** for this server.
- 4 Check **Save As Default Credentials For This Server** if you want the **ToolBar Utility** to cache the user name and password you have used to log onto that server.

Note: When the **Use Same User Name/ Password for Each Server** option is unchecked in the **ToolBar Utility**, a different user name and password can be saved for each server. (You can view the currently cached server/user associations in the expanded **ToolBar Utility** window.)

- 5 Click **OK**.

Related Concepts

[ToolBar Utility](#)

Related Procedures

[Adding Server Configurations](#)

[Opening and Closing the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

[Customizing the Toolbar Utility](#)

Logging On to the Server and Starting a Project

Before you can create a new project or open an existing project, you must select a server configuration for the project and log on.

To log on to StarTeam and create or open a project

- 1 Start the application from an icon on your desktop or by running **Start ▶ Programs ▶ StarTeam [client_name]**.
- 2 Choose **Project ▶ New** or **Project ▶ Open**.
If you have not previously logged on, this displays the **Log On to [serve_name]** dialog box.
- 3 Type a **User Name** and **Password** in the appropriate text boxes.

Note: Passwords are case sensitive and may have length restrictions.

- 4 Check **Save As Default Credentials For This Server** and click **OK**.
The New Project Wizard or the Open Project Wizard displays allowing you to create a new project or work on an existing project. On the Windows platform, logging on to StarTeam also displays the Toolbar Utility icon in your system tray. The Toolbar Utility caches a default credential for each server you access during a work session. This reduces the number of times you need to log on to the same server.
- 5 Optionally, click the Toolbar Utility icon and choose **Open** to see the list of servers for which you currently have default configurations.
- 6 Optionally, double-click the Toolbar Utility icon and choose **Use Same Name/Password** to access several server configurations with only one logon.

To log onto a StarTeam in Visual Studio and start a project

- 1 Open Visual Studio and choose **StarTeam ▶ Pull Solution**.
This displays the **Pull Solution from StarTeam** dialog box.
- 2 Choose a **StarTeam Server** from the drop-down list and click **Log On As**.
This displays the **Log On to [serve_name]** dialog box.
- 3 Type a **User Name** and **Password** in the appropriate text boxes.

Note: Passwords are case sensitive and may have length restrictions.

- 4 Choose the **Project Name** and **View Path** (if different than the default).
- 5 Browse to a different working folder if you want a different location from the one displayed.
- 6 Select the solution file to use and click **OK**.

To log on as a different user

- 1 Start the application from an icon on your desktop or choose **Start ▶ Programs ▶ StarTeam [client_name]**.
- 2 Choose **Project ▶ New** or **Project ▶ Open**.
- 3 Click the **Log On As** button.
This displays the **Log On to [serve_name]** dialog box.
- 4 Type the alternate **User Name** and **Password**.

- 5 Check **Save As Default Credentials For This Server** if you wish to reset your default credentials for this server configuration to this user name and password. This name will then appear in parentheses after the server configuration name in lists.
- 6 Click **OK** to log on.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Server Configurations](#)

[Starting Applications with the Toolbar Utility](#)

[Viewing Connection Properties](#)

[Logging Off](#)

Configuring Your Client

This section contains procedures related to configuring the client application.

In This Section

[Adding Server Configurations](#)

Describes how to add a new server configuration using the Cross-Platform Client or the Server Administration tool.

[Adding Tools to the Toolbar Utility](#)

Describes how to add more applications as tools to the Toolbar Utility.

[Changing the Default Language and Locale](#)

Describes how to change the default language and local for StarTeam.

[Changing Your Password](#)

Describes how to create and change your password for a server configuration.

[Configuring an Alternate Editor, Merge, or Comparison Utility](#)

Describes how to configure StarTeam to use an alternate editor, merge, or comparison utility

[Configuring the Display Order of Component Tabs in the Client](#)

Describes how to change the order component tabs display in the upper pane of the StarTeam client.

[Controlling How File Status Information is Stored](#)

Describes how to specify where file status information is stored.

[Customizing Personal Options](#)

Describes how set personal options to customize the application.

[Customizing the Detail Pane](#)

This topic describes how to customize the fields that display in the Detail (lower) panes of the Cross-Platform Client.

[Customizing the Toolbar Utility](#)

Describes how to customize the Toolbar Utility.

[Deleting Tools from the Toolbar Utility](#)

Describes how to delete a tool shortcut from the toolbar in the Toolbar Utility.

[Displaying Additional Fields](#)

Describes how to control which columns display in the upper pane.

[Displaying and Customizing Logging Options](#)

Describes how to customize what information is captured in the log file..

[Displaying Notifications in the Client](#)

Describes how to configure the client to display notifications in the System Tray for defects, change requests, requirements, topics, or tasks which affect you.

[Editing your Account Information](#)

Describes how to edit your user account information.

[Modifying Tool Properties in the Toolbar Utility](#)

Describes how to modify shortcut properties of a tool in the Toolbar Utility.

[Refreshing the Toolbar Utility](#)

Describes how to refresh the Toolbar Utility window after adding or deleting tools.

[Sorting and Grouping Data](#)

Describes how to sort and group data.

Adding Server Configurations

StarTeam stores all projects on the Server, which may contain numerous server configurations. You can access one or more servers from the application. However, if you have more than one server configuration running on the same computer, each server configuration must use a unique protocol and port combination. After the server is added, you can access whatever projects are available for the current server configuration of that server.

Managing server access includes adding, deleting, or modifying the server configuration properties. You can accomplish these tasks as part of creating or opening a project in one of the clients or from the Server Administration tool.

To add a new server configuration

- 1 Do one of the following:
 - ◆ Choose **Project ▶ New** or **Project ▶ Open** in the StarTeam client.
 - ◆ Choose **Start ▶ Programs ▶ Borland StarTeam ▶ StarTeam Server xxxx ▶ StarTeam Server** on a computer with a server installation. Alternatively, from the client, you can select **Start ▶ Programs ▶ Borland StarTeam ▶ StarTeam Cross-Platform Client xxxx ▶ Server Administration**. These actions display the Server Administration tool.
- 2 Do one of the following:
 - ◆ Click **Add Server** in the Cross-Platform Client **New Project** or **Open Project** wizard,
 - ◆ Choose **Server Add Server** in the Server Administration tool.
- 3 Type a unique, easy-to-remember description in the **Server description** text box.
It is not case-sensitive and can contain colons (:)
- 4 Type or browse for the computer name or IP address in the **Server address** text box.

Note: See your administrator for the server address, protocol, and endpoint information. Your administrator can also tell you what MPX profile to use if your server configuration uses StarTeamMPX.
- 5 Type the endpoint (TCP/IP port number) associated with the protocol in the **TCP/IP endpoint** text box.
- 6 Choose to specify any of the following optional settings:
 - ◆ Check the option to **Compress transferred data** if you want to use compression.
 - ◆ Select an **Encryption type** to encrypt data transferred between your workstation and the Server. Encryption protects files and other project information from being read by unauthorized parties over unsecured network lines. The encryption types are ordered (top to bottom) based on speed. Each type is slower, but safer, than the type that precedes it.
 - ◆ Click **MPX Profiles** and choose a different profile if you are using StarTeamMPX on the client and do not want to use the default profile (usually **Unicast On-site**).
- 7 Optionally, click **Properties** to review the connection properties of the selected profile.
- 8 Click **OK** when you are finished.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Files to Projects](#)

Adding Tools to the Toolbar Utility

You can add a tool to the Toolbar either by:

- ◆ Creating a new program shortcut (using the wizard that results from choosing the **Add Tool** command)
- ◆ Copying an existing program shortcut to the Toolbar Utility **Tools** folder.

Note: After adding a tool to the **Toolbar Utility**, you must choose the **Refresh Tools** command to refresh the toolbar region of the window and the context menu for the system tray icon. (You might also want to modify the properties of the tool.)

If you add so many tools to the **Toolbar Utility** that they cannot be displayed at the window's current size, you should enlarge the window so all the tool icons are visible.

To add a tool by creating a new shortcut

- 1 Right-click the toolbar region of the **Toolbar Utility** window and choose **Add Tool**.
- 2 On the **Create Shortcut** page of the resulting wizard, specify the command line for starting the tool, and click **Next**.
If you do not know the exact location of the executable file for the program, then click **Browse** to open a dialog box that enables you to locate and select the necessary file.
- 3 On the **Select a Title for the Program** page of the wizard, specify the name for the tool shortcut and click **Finish**.
The specified program shortcut is created in the **Toolbar Utility Tools** folder.
- 4 Right-click the toolbar region of the **Toolbar Utility** window and choose **Refresh Tools**.
The tool icon is added to the toolbar region of the window, and the shortcut name is added to the context menu for the **System Tray** icon.

To add a tool by copying an existing shortcut

- 1 Right-click the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**.
In Windows, the resulting Windows Explorer window shows the contents of that folder, which should consist of the program shortcuts for the tools on the toolbar.
- 2 In Windows Explorer, locate an existing program shortcut. (On Windows NT, search for the **Shortcut** file type. On later editions of Windows, search for files named ***.lnk**)
- 3 Copy the existing program shortcut to the **Toolbar Utility Tools** folder.
- 4 Right-click the toolbar region of the **Toolbar Utility** window and choose **Refresh Tools**.
The tool icon is added to the toolbar region of the window, and the shortcut name is added to the context menu for the **System Tray** icon.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Changing the Default Language and Locale

You can change the default language and local for the application by modifying the `StarTeamCP.stjava` file in the root StarTeam directory.

To change the default language and local for the application

- 1 Open the `StarTeamCP.stjava` text file in any text editor.
- 2 Edit the line `options=-Xmx256m` to be one of the following:

Language	Line Contents
German	<code>options=-Xmx256m -Duser.language=de -Duser.country=DE</code>
English	<code>options=-Xmx256m -Duser.language=en -Duser.country=EN</code>
French	<code>options=-Xmx256m -Duser.language=fr -Duser.country=FR</code>
Japanese	<code>options=-Xmx256m -Duser.language=ja -Duser.country=JA</code>

Related Concepts

[Personal Options](#)

Changing Your Password

Occasionally, you might need to change your password. For example, you might need to change it every three months to follow company policy.

To change your password

- 1 Choose **Tools** ► **My Account** to open the **My Account** dialog box.
- 2 Click the **Logon** tab.
- 3 Do one of the following:
 - ◆ Type your new password in the **Password** and **Confirm** text boxes.
 - ◆ Check **Set a Blank Password** if your administrator allows blank passwords and you want to use one.
- 4 Click **OK**.

Related Concepts

[Personal Options](#)

Related Procedures

[Adding Server Configurations](#)

[Editing your Account Information](#)

[Displaying and Customizing Logging Options](#)

Configuring an Alternate Editor, Merge, or Comparison Utility

You can specify an alternate editor, merge utility, and comparison utility to use in the application if you don't want to use the default tools for those functions.

To use an alternate editor, merge, or comparison utility

- 1 Choose **Tools** ► **Personal Options** and select the **File** tab.
- 2 Check one or more types of alternate applications you want to use: **Editor**, **Merge Utility**, and **Comparison Utility**.
- 3 Type or browse to the path for executable file for each selected application.
- 4 Type any options you want to use in the **Options** text boxes for the selected applications.
- 5 Click **OK** when finished.

Note: For non-Windows systems, specify a command to use for launching files in an alternate application in the **Open with...** text box on the **File** or **Folder** tab in the **Personal Options** dialog box.

Related Concepts

[Personal Options](#)

Related Procedures

[Customizing Personal Options](#)

Related Reference

[File Options \(Personal Options Dialog Box\)](#)

Configuring the Display Order of Component Tabs in the Client

This procedure describes how to change the order component tabs display in the upper pane of the StarTeam client.

Note: When reordering the tabs in the upper pane, do not make the **Audit** tab the first tab. Listing all the audit entries can take a very long time.

To change the order component tabs display in the client

- 1 Choose **Tools** ► **Personal Options**.
- 2 On the **Workspace** tab , click the **Component Order** button to the right of the component tab display.
This opens the **Select Component Order** dialog box which displays two lists of components: On the left, available components to add to the display, and on the right, the current components in the order they are being displayed.
- 3 Do one of the following:
 - ◆ To remove a component from the current display, select it on the right and click **Remove**.
 - ◆ To add a component to the current display, select it on the left and click **Add**. The component is added to the end of the list.
 - ◆ To change the order of the component tabs, remove them all from the list on the right, then add them back in the desired display order.
- 4 Click **OK** to close the **Select Component Order** dialog box, then **OK** again to close the **Personal Options** dialog box.

Note: You must close and reopen your project to see the changes.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)
[StarTeamMPX Options \(Personal Options Dialog Box\)](#)
[File Options \(Personal Options Dialog Box\)](#)
[Change Request Options \(Personal Options Dialog Box\)](#)
[Requirements Options \(Personal Options Dialog Box\)](#)
[Task Options \(Personal Options Dialog Box\)](#)
[Topic Options \(Personal Options Dialog Box\)](#)

Controlling How File Status Information is Stored

File status information about the files you are working on is stored on your workstation either in a central location or in a child folder (named `.sbas`) of each working folder.

You can set the file status property for a specific view. The view property defaults to the storage method that you selected as a personal option. When changed from that default, the view property take precedence over your personal option for the view.

Note: You can also set your Personal Options to control file status information for all your files, unless those files are in views for which you have set the view property for file status.

To set the file status property for a view

- 1 Choose **View** ► **Properties** to open the **View Properties** dialog box, then select the **Name** page.
- 2 Select the **Central** or **Per Folder** option button in the **File Status Repository** group box.

The per-folder option is most useful in the special case where multiple users are sharing a working folder; for example, on a shared network drive.

For example, suppose several users all check files in and out of a shared working folder. If these users have set the central repository option for file statuses, the statuses are stored on each of their computers. Whenever a user makes a change to a file in the working folder, the status for that file is undated only on that user's computer. Everyone else sees the status **Unknown** for that file. Over time, all the files may have been changed, and the statuses can become **Unknown** for all users of all files. Using the per-folder option causes the statuses to be updated within the working folder itself. Everyone has access to those status changes and **Unknown** statuses do not occur.

- 3 Click **OK**.

Tip: Select **Default (Central)** to return to using the **Personal Options** settings.

Related Concepts

[Personal Options](#)
[Proper Use of Views](#)

Related Procedures

[Customizing Personal Options](#)
[Reviewing or Modifying View Properties](#)

Related Reference

[File Status Information](#)
[Effects of Status on Check-ins and Check-outs](#)

Customizing Personal Options

Personal Options allow you to customize the application by adjusting the way the following elements work:

- ◆ Your **Workspace**: Options include action confirmation, display/hide application tools, folder selection, screen refreshes, report path, and client log preferences.
- ◆ **StarTeamMPX** server: Options include application enablement and refresh times.
- ◆ **File** component: Options include general preferences, check-out, locking, and merging options, repository preferences, use of alternate editors, and so on.
- ◆ **Change Request** component: Options include marking change requests when selected or read, system tray notification, and locking.
- ◆ **Requirement** component: Options include marking requirements when selected or read, system tray notification, and locking.
- ◆ **Task** component: Options include marking tasks when selected or read, system tray notification, and locking.
- ◆ **Topic** component: Options include marking topics when selected or read, system tray notification, and locking.

To customize your personal options

- 1 Choose **Tools** ► **Personal Options**.
- 2 Select the tab which contains the options you want to change.
- 3 Make the desired changes, then click **OK**.

Note: It is necessary to close and reopen the project for component tab order changes to take effect.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Configuring the Display Order of Component Tabs in the Client](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

Customizing the Detail Pane

You can modify the display format and content of the **Detail** (lower) pane in the Cross-Platform Client on a per-workstation basis by placing correctly named and formatted HTML templates in the same directory as the `starteam-servers.xml` and `starteam-client-options.xml` files. For example, on a Windows system these files could be located in the `C:\Documents and Settings\USER\Application Data\Borland\StarTeam` folder.

Sample **Detail** pane templates are installed under the Cross-Platform Client root installation folder in the `samples\details-templates` folder.

The name of the template file controls the StarTeam component to be modified. For example, a template named `changerequest.details.html` controls the format and content of the **Detail** pane display for the change request component.

To customize the detail panes on your workstation

- 1 Create an HTML template file for the corresponding component that you wish to customize. For example, if you want to format the contents of the **Detail** pane for a change request, you would create a template file named `changerequest.details.html`.

You must use the following file names for the component detail panes that you wish to customize:

- ◆ `folder.details.html`
- ◆ `file.details.html`
- ◆ `changerequest.details.html`
- ◆ `task.details.html`
- ◆ `topic.details.html`
- ◆ `requirement.details.html`
- ◆ `changepackage.details.html`

Tip: Refer to the link at the end of this topic to review sample template files that you can use as a starting point for creating your own templates for customizing the **Detail** pane.

- 2 Make any desired modifications to the template file.
Follow the formatting example in the sample template file. The fields used in the **Detail** pane HTML templates are recognized by the client when they are contained between double tilde `~~` characters. For example: `~~Status~~` represents the Status field found in the **Change Request Properties** dialog box.
- 3 Save the template files in the same directory as the `starteam-servers.xml` and `starteam-client-options.xml` files. For example, on a Windows system these files could be located in the `C:\Documents and Settings\USER\Application Data\Borland\StarTeam` folder.

Related Reference

[Detail Pane Customization Reference](#)

Customizing the Toolbar Utility

The toolbar in the **Toolbar Utility** is initially populated with shortcuts for the tools of the products that are installed on your workstation. If you want to launch other programs from the **Toolbar Utility**, you must add program shortcuts to the toolbar. The tools appear on the toolbar in alphabetical order, based on the name of their program shortcuts.

You can customize the toolbar either by:

- ◆ Using the commands on the **Toolbar Utility** context menu, which is accessed by right-clicking on the toolbar region.
- ◆ Working directly with the program shortcuts in the **Toolbar Utility Tools** folder using an Explorer window.

Note: After adding, modifying, or deleting a tool, you must use the **Refresh Tools** command so the changes will be reflected in the toolbar region of the window and on the context menu for the **System Tray** icon.

To customize the Toolbar Utility, do one of the following

- 1 Add applications to the **Toolbar Utility** as tools on its toolbar.

[Adding Tools to the Toolbar Utility](#)

- 2 Modify the properties of a tool.

[Modifying Tool Properties in the Toolbar Utility](#)

- 3 Delete a tool.

[Deleting Tools from the Toolbar Utility](#)

- 4 Refresh the **Toolbar Utility** window.

[Refreshing the Toolbar Utility](#)

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Adding Tools to the Toolbar Utility

You can add a tool to the Toolbar either by:

- ◆ Creating a new program shortcut (using the wizard that results from choosing the **Add Tool** command)
- ◆ Copying an existing program shortcut to the Toolbar Utility **Tools** folder.

Note: After adding a tool to the **Toolbar Utility**, you must choose the **Refresh Tools** command to refresh the toolbar region of the window and the context menu for the system tray icon. (You might also want to modify the properties of the tool.)

If you add so many tools to the **Toolbar Utility** that they cannot be displayed at the window's current size, you should enlarge the window so all the tool icons are visible.

To add a tool by creating a new shortcut

- 1 Right-click the toolbar region of the **Toolbar Utility** window and choose **Add Tool**.
- 2 On the **Create Shortcut** page of the resulting wizard, specify the command line for starting the tool, and click **Next**.

If you do not know the exact location of the executable file for the program, then click **Browse** to open a dialog box that enables you to locate and select the necessary file.
- 3 On the **Select a Title for the Program** page of the wizard, specify the name for the tool shortcut and click **Finish**.

The specified program shortcut is created in the **Toolbar Utility Tools** folder.
- 4 Right-click the toolbar region of the **Toolbar Utility** window and choose **Refresh Tools**.

The tool icon is added to the toolbar region of the window, and the shortcut name is added to the context menu for the **System Tray** icon.

To add a tool by copying an existing shortcut

- 1 Right-click the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**.

In Windows, the resulting Windows Explorer window shows the contents of that folder, which should consist of the program shortcuts for the tools on the toolbar.
- 2 In Windows Explorer, locate an existing program shortcut. (On Windows NT, search for the **Shortcut** file type. On later editions of Windows, search for files named ***.lnk**)
- 3 Copy the existing program shortcut to the **Toolbar Utility Tools** folder.
- 4 Right-click the toolbar region of the **Toolbar Utility** window and choose **Refresh Tools**.

The tool icon is added to the toolbar region of the window, and the shortcut name is added to the context menu for the **System Tray** icon.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Modifying Tool Properties in the Toolbar Utility

You might want to modify the shortcut properties of a tool in the **Toolbar Utility**. For example, you might want to rename a tool to include the program version number. Or, you might have accepted the default name while creating the shortcut, and you want to change it from the name of the executable file to something more descriptive.

Modifying the properties of a shortcut in the **Toolbar Utility** is much like changing the properties for a desktop shortcut. Both use the same **Properties** dialog box.

Note: The **Properties** dialog box in Windows NT does not allow you to change the name of the shortcut. However, you can rename a shortcut from an Explorer window (by editing the shortcut name), as explained in the second procedure below.

To modify the tool properties

- 1 Do one of the following to open the **Properties** dialog box.
 - ◆ Right-click a tool icon on the toolbar in the **Toolbar Utility**, and choose **Modify Tool**.
 - ◆ Right-click on the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**. In the resulting Explorer window, right-click on the program shortcut for the tool and choose **Properties**.
- 2 Change the properties for the shortcut, then click **OK** when you are finished.

To rename a tool in Windows NT

- 1 Right-click on the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**.
- 2 Do either of the following in the resulting Explorer window:
 - ◆ Right-click the program shortcut for the tool and choose **Rename**.
 - ◆ Click on the program shortcut for the tool, and click again on the text below the icon.
- 3 Edit the name, and press the **TYPE** key.
- 4 Right-click on the toolbar region of the **Toolbar Utilities** window and choose **Refresh Tools**.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Deleting Tools from the Toolbar Utility

If you need to delete a tool in the **Toolbar Utility** , you can do so from either the **Toolbar Utility** window (using its **Delete Tool** command) or an Explorer window.

To delete a tool from the Toolbar Utility

- 1 Right-click the tool icon on the toolbar and choose **Delete Tool**.
- 2 Click **Yes** in the confirmation dialog box.
The toolbar is automatically refreshed.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Refreshing the Toolbar Utility

After adding, modifying, or deleting a shortcut for a tool in the **Toolbar Utility**, you must refresh the toolbar so that it reflects your changes. This action is necessary regardless of whether you used the commands on the **Toolbar Utility** context menu, or worked directly with the shortcuts in the **Tools** folder. (The only exception is deleting a tool using the **Delete Tool** command, which automatically refreshes the toolbar.)

To refresh the toolbar in the Toolbar Utility window

- 1 Right-click the toolbar region of the **Toolbar Utility** window.
- 2 Choose **Refresh Tools**.

This command refreshes the toolbar region of the window and the context menu for the **System Tray** icon by processing all the program shortcuts in the **Tools** folder.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Deleting Tools from the Toolbar Utility

If you need to delete a tool in the **Toolbar Utility** , you can do so from either the **Toolbar Utility** window (using its **Delete Tool** command) or an Explorer window.

To delete a tool from the Toolbar Utility

- 1 Right-click the tool icon on the toolbar and choose **Delete Tool**.
- 2 Click **Yes** in the confirmation dialog box.
The toolbar is automatically refreshed.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Displaying Additional Fields

You can change what column headers appear in the upper pane or in the History tab. These headers can be hidden, displayed, or re-ordered, and in the upper pane, affect what you can find using the [Find](#) command.

To change column headers

1 Do one of the following:

- ◆ Right-click a column header on the upper pane or in the History tab. Then choose [Show Fields](#) from the context menu.
- ◆ Choose [Filters](#) ► [Show Fields](#) from the appropriate item menu or context menu. For example, if files are displayed in the upper pane, choose this option from the [File](#) menu.

The **Show Fields** dialog box displays two lists. The **Available Fields** list box contains all the fields that could be displayed as column headers but are not currently displayed. The **Show These Fields in This Order** list box displays all the fields that are currently displayed.

2 Do any combination of the following:

- ◆ Display additional fields in the upper pane or in the History tab. To do this, select the fields to appear as the column headers from the Available fields list. Then click Add.
- ◆ Stop displaying fields in the upper pane or in the History tab. To do this, select the fields to be removed from the Show these fields in this order list box. Then click Remove.
- ◆ To change the order of the fields to be displayed in the upper pane or in the History tab, drag each field name to the desired location in the Show these fields in this order list box.

3 Click **OK** when you are finished.

Tip: Double-clicking a field name moves it from one list box to the other. The **Show Fields** dialog box initially displays the most commonly used fields. Check the **Show Advanced Fields** check box to select from a complete list of the available fields.

Related Concepts

[Cross-Platform Client Overview](#)
[Sorting and Grouping Data](#)

Related Procedures

[Filtering Data](#)
[Creating Filters](#)
[Creating Queries](#)
[Creating Reports](#)

Displaying and Customizing Logging Options

The [StarTeam.Log](#) file records the operations performed on your workstation during a work session. Reviewing logs can help you or your administrator troubleshoot errors or failed operations.

StarTeam.log contains data about operations sent from your workstation to one or more servers, depending on what project views you have open. This data includes the name of the project so that you can isolate data for a particular server, when necessary. Depending upon the selections made in **Personal Options** dialog box, your [StarTeam.Log](#) file can record the following types of information: error messages, operation summaries, and details about the individual commands required to perform each operation.

To view the contents of the [StarTeam.Log](#) file, do one of the following

- ◆ Choose **Tools** ► **StarTeam Log** in the StarTeam client.
- ◆ Import and view the data from a [StarTeam.Log](#) file using any application that supports tab-delimited fields. For example, if you save the file with a [.csv](#) extension, you can open the file in Microsoft Excel.

To customize the StarTeam.Log file

- 1 Choose **Tools** ► **PersonalOptions** and click the **Workspace** tab in the **Personal Options** dialog box.
- 2 Type or browse for the location of the [StarTeam.Log](#) file in the **Log Output Path** text box.
The default is the location in which the application is installed; for example, [C:\Program Files\Borland\StarTeam client_name\Log](#).
The current log file is always named [StarTeam.log](#). Log files from earlier sessions of the application include the date and time the file was last modified.
- 3 Select the types of data you want to include in [StarTeam.Log](#).
- 4 Click **OK** when you are finished.

Related Concepts

[Personal Options](#)

Related Procedures

[Customizing Personal Options](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)
[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

Displaying Notifications in the Client

Some personal options control how often you are alerted about new items using **System Tray** notification. In the StarTeam, notification icons appear on the **Status Bar**.

While you are running the application, you can check for changes in items that may affect you. This feature does not apply to files and audit entries. The application notifies you when:

- ◆ A change request, requirement, or task becomes your responsibility or a topic names you as a recipient. If a topic has no recipients listed, no one receives notification.
- ◆ A requirement or task that is your responsibility or a topic for which you are a recipient has changed.

Note: The defect, requirement, task, or topic icon display at the right end of the **Status Bar**.

To enable System Tray notification

- 1 Choose **Tools** ► **Personal Options** to open the **Personal Options** dialog box.
- 2 Select the **Change Request**, **Requirement**, **Topic**, or **Task** tab.
- 3 Check **Check for New or Modified [item]** in the **System Tray Notifications** group box.
- 4 Type the number of minutes for the time interval between checks for items that need your attention. The default is 10.
The dialog box displays the icon that will appear in the **System Tray** for that particular item.
- 5 Click **OK** to save your options and close the **Personal Options** dialog box.

To open a System Tray notification item

- 1 Double-click the defect, requirement, task, or topic notification icon in the **System Tray** to open the **[Project] New [Item Type]** dialog box which lists the items that need your attention.
- 2 Double-click the item to display its properties.
You might want to take notes while displaying the item properties because as soon as you close the dialog box the item disappears.

Note: If your administrator has enabled email notification, you will automatically receive email messages notifying you about change requests for which you are responsible, about changes in any requirements and tasks for which you are responsible, and about changes in any topics for which you are a recipient. Email notification is client independent and you do not need to run the application to receive notifications. You can, however, use the **System Tray** notification with or without email notification.

Related Concepts

[Personal Options](#)

Related Procedures

[Customizing Personal Options](#)

[Controlling How File Status Information is Stored](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

Editing your Account Information

Occasionally, you must update information about your user account information on the server, such as your name, contact information, and password.

To edit your account information

- 1 Choose **Tools** ► **MyAccount** to open the **My Account** dialog box.
- 2 On the **General** tab, enter any missing information or change any incorrect information in the fields available.
- 3 Click **OK**.

Related Concepts

[Personal Options](#)

Related Procedures

[Adding Server Configurations](#)

[Changing Your Password](#)

[Displaying and Customizing Logging Options](#)

Modifying Tool Properties in the Toolbar Utility

You might want to modify the shortcut properties of a tool in the **Toolbar Utility**. For example, you might want to rename a tool to include the program version number. Or, you might have accepted the default name while creating the shortcut, and you want to change it from the name of the executable file to something more descriptive.

Modifying the properties of a shortcut in the **Toolbar Utility** is much like changing the properties for a desktop shortcut. Both use the same **Properties** dialog box.

Note: The **Properties** dialog box in Windows NT does not allow you to change the name of the shortcut. However, you can rename a shortcut from an Explorer window (by editing the shortcut name), as explained in the second procedure below.

To modify the tool properties

- 1 Do one of the following to open the **Properties** dialog box.
 - ◆ Right-click a tool icon on the toolbar in the **Toolbar Utility**, and choose **Modify Tool**.
 - ◆ Right-click on the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**. In the resulting Explorer window, right-click on the program shortcut for the tool and choose **Properties**.
- 2 Change the properties for the shortcut, then click **OK** when you are finished.

To rename a tool in Windows NT

- 1 Right-click on the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**.
- 2 Do either of the following in the resulting Explorer window:
 - ◆ Right-click the program shortcut for the tool and choose **Rename**.
 - ◆ Click on the program shortcut for the tool, and click again on the text below the icon.
- 3 Edit the name, and press the **TYPE** key.
- 4 Right-click on the toolbar region of the **Toolbar Utilities** window and choose **Refresh Tools**.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Refreshing the Toolbar Utility

After adding, modifying, or deleting a shortcut for a tool in the **Toolbar Utility**, you must refresh the toolbar so that it reflects your changes. This action is necessary regardless of whether you used the commands on the **Toolbar Utility** context menu, or worked directly with the shortcuts in the **Tools** folder. (The only exception is deleting a tool using the **Delete Tool** command, which automatically refreshes the toolbar.)

To refresh the toolbar in the Toolbar Utility window

- 1 Right-click the toolbar region of the **Toolbar Utility** window.
- 2 Choose **Refresh Tools**.

This command refreshes the toolbar region of the window and the context menu for the **System Tray** icon by processing all the program shortcuts in the **Tools** folder.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Sorting and Grouping Data

You can choose to do a primary sort in the upper pane (based on one column), or a more complicated sorts up to a fourth order.

To do a primary sort on one column

- 1 Open the view on the data you wish to sort or group.
- 2 Click a column header to sort the data in the upper pane based on the value in that column.
The sort is in ascending order by number, letter, internal order, or internal key, depending on the data.
- 3 Click the column header again to reverse the sort order.

A triangle appears on column header of the sorted column. The triangle points upward for ascending sorts and downward for descending sorts.

Note: You can also sort the data in the lower pane when the **Link** tab is selected.

To perform up to a fourth-order sort

- 1 Do one of the following:
 - ◆ Right-click a column header on upper pane and choose **Sort and Group** from the context menu.
 - ◆ Right-click in the upper pane and choose **Filters** ► **Sort and Group** from the context menu.

The **Sort and Group** dialog box displays four group boxes, each indented slightly more to the right than the one above it. The first group box designates a primary sort order, the second designates a secondary sort, and so on.

- 2 Optionally, check the **Show Advanced Fields** check box at the bottom of the dialog box to list all the fields in **First By** and **Then By** drop-down list boxes. Some fields are rarely used and considered advanced.
- 3 Select a field from the **First By** drop-down list box.
If you are grouping the items, the field does not need to be displayed in the upper pane. If you are not grouping the items, you can sort them based on a field that is not displayed, but you will not be able to tell where one group leaves off and the next begins.
- 4 Select the **Ascending** or **Descending** option button. The default setting is ascending order.
- 5 Select **Group By** to group the items which have the same values in this field.
If you do not select any additional sort options, text fields are sorted in ASCII order. Enumerated and user ID fields are sorted by their internal order or internal keys. That is, enumerated fields are sorted in the order given to them by the person who created the field; user ID fields are sorted in the order in which they were created. The application disables the Sort Options button for numeric and date/time fields.
- 6 Optionally, click **Sort Options** for additional sorting selections. The **Sort Options** dialog box appears.
 - ◆ Select **As Text** to sort enumerated and user ID fields by the names of their possible values. For text fields, **As Text** is your only choice.
 - ◆ Uncheck the **Case-sensitive** check box to sort alphabetically or check it to sort in ASCII order (where uppercase letters precede lowercase letters).
- 7 Add secondary and lower order sorts by using the **Then By Group** boxes as needed.

Related Concepts

[Cross-Platform Client Overview](#)

[Displaying Additional Fields](#)

Related Procedures

[Filtering Data](#)

[Creating Filters](#)

[Creating Queries](#)

[Creating Reports](#)

Using the Toolbar Utility

This section contains procedures related to using and configuring the Toolbar Utility.

In This Section

[Adding Tools to the Toolbar Utility](#)

Describes how to add more applications as tools to the Toolbar Utility.

[Customizing the Toolbar Utility](#)

Describes how to customize the Toolbar Utility.

[Deleting Tools from the Toolbar Utility](#)

Describes how to delete a tool shortcut from the toolbar in the Toolbar Utility.

[Logging On to Servers with the Toolbar Utility](#)

Describes how to log on to a server.

[Modifying Tool Properties in the Toolbar Utility](#)

Describes how to modify shortcut properties of a tool in the Toolbar Utility.

[Opening and Closing the Toolbar Utility](#)

Describes how to open and close the Toolbar Utility

[Refreshing the Toolbar Utility](#)

Describes how to refresh the Toolbar Utility window after adding or deleting tools.

[Starting Applications with the Toolbar Utility](#)

Describes how to launch applications from the Toolbar Utility.

Adding Tools to the Toolbar Utility

You can add a tool to the Toolbar either by:

- ◆ Creating a new program shortcut (using the wizard that results from choosing the **Add Tool** command)
- ◆ Copying an existing program shortcut to the Toolbar Utility **Tools** folder.

Note: After adding a tool to the **Toolbar Utility**, you must choose the **Refresh Tools** command to refresh the toolbar region of the window and the context menu for the system tray icon. (You might also want to modify the properties of the tool.)

If you add so many tools to the **Toolbar Utility** that they cannot be displayed at the window's current size, you should enlarge the window so all the tool icons are visible.

To add a tool by creating a new shortcut

- 1 Right-click the toolbar region of the **Toolbar Utility** window and choose **Add Tool**.
- 2 On the **Create Shortcut** page of the resulting wizard, specify the command line for starting the tool, and click **Next**.
If you do not know the exact location of the executable file for the program, then click **Browse** to open a dialog box that enables you to locate and select the necessary file.
- 3 On the **Select a Title for the Program** page of the wizard, specify the name for the tool shortcut and click **Finish**.
The specified program shortcut is created in the **Toolbar Utility Tools** folder.
- 4 Right-click the toolbar region of the **Toolbar Utility** window and choose **Refresh Tools**.
The tool icon is added to the toolbar region of the window, and the shortcut name is added to the context menu for the **System Tray** icon.

To add a tool by copying an existing shortcut

- 1 Right-click the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**.
In Windows, the resulting Windows Explorer window shows the contents of that folder, which should consist of the program shortcuts for the tools on the toolbar.
- 2 In Windows Explorer, locate an existing program shortcut. (On Windows NT, search for the **Shortcut** file type. On later editions of Windows, search for files named ***.lnk**)
- 3 Copy the existing program shortcut to the **Toolbar Utility Tools** folder.
- 4 Right-click the toolbar region of the **Toolbar Utility** window and choose **Refresh Tools**.
The tool icon is added to the toolbar region of the window, and the shortcut name is added to the context menu for the **System Tray** icon.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Customizing the Toolbar Utility

The toolbar in the **Toolbar Utility** is initially populated with shortcuts for the tools of the products that are installed on your workstation. If you want to launch other programs from the **Toolbar Utility**, you must add program shortcuts to the toolbar. The tools appear on the toolbar in alphabetical order, based on the name of their program shortcuts.

You can customize the toolbar either by:

- ◆ Using the commands on the **Toolbar Utility** context menu, which is accessed by right-clicking on the toolbar region.
- ◆ Working directly with the program shortcuts in the **Toolbar Utility Tools** folder using an Explorer window.

Note: After adding, modifying, or deleting a tool, you must use the **Refresh Tools** command so the changes will be reflected in the toolbar region of the window and on the context menu for the **System Tray** icon.

To customize the Toolbar Utility, do one of the following

- 1 Add applications to the **Toolbar Utility** as tools on its toolbar.

[Adding Tools to the Toolbar Utility](#)

- 2 Modify the properties of a tool.

[Modifying Tool Properties in the Toolbar Utility](#)

- 3 Delete a tool.

[Deleting Tools from the Toolbar Utility](#)

- 4 Refresh the **Toolbar Utility** window.

[Refreshing the Toolbar Utility](#)

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Adding Tools to the Toolbar Utility

You can add a tool to the Toolbar either by:

- ◆ Creating a new program shortcut (using the wizard that results from choosing the **Add Tool** command)
- ◆ Copying an existing program shortcut to the Toolbar Utility **Tools** folder.

Note: After adding a tool to the **Toolbar Utility**, you must choose the **Refresh Tools** command to refresh the toolbar region of the window and the context menu for the system tray icon. (You might also want to modify the properties of the tool.)

If you add so many tools to the **Toolbar Utility** that they cannot be displayed at the window's current size, you should enlarge the window so all the tool icons are visible.

To add a tool by creating a new shortcut

- 1 Right-click the toolbar region of the **Toolbar Utility** window and choose **Add Tool**.
- 2 On the **Create Shortcut** page of the resulting wizard, specify the command line for starting the tool, and click **Next**.

If you do not know the exact location of the executable file for the program, then click **Browse** to open a dialog box that enables you to locate and select the necessary file.
- 3 On the **Select a Title for the Program** page of the wizard, specify the name for the tool shortcut and click **Finish**.

The specified program shortcut is created in the **Toolbar Utility Tools** folder.
- 4 Right-click the toolbar region of the **Toolbar Utility** window and choose **Refresh Tools**.

The tool icon is added to the toolbar region of the window, and the shortcut name is added to the context menu for the **System Tray** icon.

To add a tool by copying an existing shortcut

- 1 Right-click the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**.

In Windows, the resulting Windows Explorer window shows the contents of that folder, which should consist of the program shortcuts for the tools on the toolbar.
- 2 In Windows Explorer, locate an existing program shortcut. (On Windows NT, search for the **Shortcut** file type. On later editions of Windows, search for files named ***.lnk**)
- 3 Copy the existing program shortcut to the **Toolbar Utility Tools** folder.
- 4 Right-click the toolbar region of the **Toolbar Utility** window and choose **Refresh Tools**.

The tool icon is added to the toolbar region of the window, and the shortcut name is added to the context menu for the **System Tray** icon.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Modifying Tool Properties in the Toolbar Utility

You might want to modify the shortcut properties of a tool in the **Toolbar Utility**. For example, you might want to rename a tool to include the program version number. Or, you might have accepted the default name while creating the shortcut, and you want to change it from the name of the executable file to something more descriptive.

Modifying the properties of a shortcut in the **Toolbar Utility** is much like changing the properties for a desktop shortcut. Both use the same **Properties** dialog box.

Note: The **Properties** dialog box in Windows NT does not allow you to change the name of the shortcut. However, you can rename a shortcut from an Explorer window (by editing the shortcut name), as explained in the second procedure below.

To modify the tool properties

- 1 Do one of the following to open the **Properties** dialog box.
 - ◆ Right-click a tool icon on the toolbar in the **Toolbar Utility**, and choose **Modify Tool**.
 - ◆ Right-click on the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**. In the resulting Explorer window, right-click on the program shortcut for the tool and choose **Properties**.
- 2 Change the properties for the shortcut, then click **OK** when you are finished.

To rename a tool in Windows NT

- 1 Right-click on the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**.
- 2 Do either of the following in the resulting Explorer window:
 - ◆ Right-click the program shortcut for the tool and choose **Rename**.
 - ◆ Click on the program shortcut for the tool, and click again on the text below the icon.
- 3 Edit the name, and press the **TYPE** key.
- 4 Right-click on the toolbar region of the **Toolbar Utilities** window and choose **Refresh Tools**.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Deleting Tools from the Toolbar Utility

If you need to delete a tool in the **Toolbar Utility** , you can do so from either the **Toolbar Utility** window (using its **Delete Tool** command) or an Explorer window.

To delete a tool from the Toolbar Utility

- 1 Right-click the tool icon on the toolbar and choose **Delete Tool**.
- 2 Click **Yes** in the confirmation dialog box.
The toolbar is automatically refreshed.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Refreshing the Toolbar Utility

After adding, modifying, or deleting a shortcut for a tool in the **Toolbar Utility**, you must refresh the toolbar so that it reflects your changes. This action is necessary regardless of whether you used the commands on the **Toolbar Utility** context menu, or worked directly with the shortcuts in the **Tools** folder. (The only exception is deleting a tool using the **Delete Tool** command, which automatically refreshes the toolbar.)

To refresh the toolbar in the Toolbar Utility window

- 1 Right-click the toolbar region of the **Toolbar Utility** window.
- 2 Choose **Refresh Tools**.

This command refreshes the toolbar region of the window and the context menu for the **System Tray** icon by processing all the program shortcuts in the **Tools** folder.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Deleting Tools from the Toolbar Utility

If you need to delete a tool in the **Toolbar Utility** , you can do so from either the **Toolbar Utility** window (using its **Delete Tool** command) or an Explorer window.

To delete a tool from the Toolbar Utility

- 1 Right-click the tool icon on the toolbar and choose **Delete Tool**.
- 2 Click **Yes** in the confirmation dialog box.
The toolbar is automatically refreshed.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Logging On to Servers with the Toolbar Utility

This procedure describes the steps for logging onto a server in StarTeam.

Log on to a server

- 1 Choose **Project** ► **Open** to launch the **Open Project Wizard**.
- 2 Select the server in the **Server/Project Tree** and click the **Log On As** button to open the **Log On To** dialog box:

Note: If the server you want is not in the list, click **Add Server** to add a new server configuration.

- 3 Type your **User Name** and **Password** for this server.
- 4 Check **Save As Default Credentials For This Server** if you want the **ToolBar Utility** to cache the user name and password you have used to log onto that server.

Note: When the **Use Same User Name/ Password for Each Server** option is unchecked in the **ToolBar Utility**, a different user name and password can be saved for each server. (You can view the currently cached server/user associations in the expanded **ToolBar Utility** window.)

- 5 Click **OK**.

Related Concepts

[ToolBar Utility](#)

Related Procedures

[Adding Server Configurations](#)

[Opening and Closing the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

[Customizing the Toolbar Utility](#)

Modifying Tool Properties in the Toolbar Utility

You might want to modify the shortcut properties of a tool in the **Toolbar Utility**. For example, you might want to rename a tool to include the program version number. Or, you might have accepted the default name while creating the shortcut, and you want to change it from the name of the executable file to something more descriptive.

Modifying the properties of a shortcut in the **Toolbar Utility** is much like changing the properties for a desktop shortcut. Both use the same **Properties** dialog box.

Note: The **Properties** dialog box in Windows NT does not allow you to change the name of the shortcut. However, you can rename a shortcut from an Explorer window (by editing the shortcut name), as explained in the second procedure below.

To modify the tool properties

- 1 Do one of the following to open the **Properties** dialog box.
 - ◆ Right-click a tool icon on the toolbar in the **Toolbar Utility**, and choose **Modify Tool**.
 - ◆ Right-click on the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**. In the resulting Explorer window, right-click on the program shortcut for the tool and choose **Properties**.
- 2 Change the properties for the shortcut, then click **OK** when you are finished.

To rename a tool in Windows NT

- 1 Right-click on the toolbar region of the **Toolbar Utility** window and choose **Open Tools Folder**.
- 2 Do either of the following in the resulting Explorer window:
 - ◆ Right-click the program shortcut for the tool and choose **Rename**.
 - ◆ Click on the program shortcut for the tool, and click again on the text below the icon.
- 3 Edit the name, and press the **TYPE** key.
- 4 Right-click on the toolbar region of the **Toolbar Utilities** window and choose **Refresh Tools**.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Opening and Closing the Toolbar Utility

The **Toolbar Utility** is started automatically by the clients and is minimized as an icon in the **System Tray**.

To manually open and close the Toolbar Utility

1 Open the **Toolbar Utility** by doing one of the following:

- ◆ Double-click the **Toolbar Utility** icon on the **System Tray**.
- ◆ Right-click the **Toolbar Utility** icon on the **System Tray** and choose **Open**.

Note: When no portion of the **Toolbar Utility** user interface is visible, you can display its expanded window by starting a second instance of it. To do so, choose **Start ▶ Programs ▶ StarTeam ▶ StarTeam Toolbar**.

2 Close the **Toolbar Utility** by doing one of the following:

- ◆ Right-click the **Toolbar Utility** icon in the **System Tray** and choose **Exit**.
- ◆ Click the **Close** icon on the **Toolbar Utility** title bar.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

[Customizing the Toolbar Utility](#)

Refreshing the Toolbar Utility

After adding, modifying, or deleting a shortcut for a tool in the **Toolbar Utility**, you must refresh the toolbar so that it reflects your changes. This action is necessary regardless of whether you used the commands on the **Toolbar Utility** context menu, or worked directly with the shortcuts in the **Tools** folder. (The only exception is deleting a tool using the **Delete Tool** command, which automatically refreshes the toolbar.)

To refresh the toolbar in the Toolbar Utility window

- 1 Right-click the toolbar region of the **Toolbar Utility** window.
- 2 Choose **Refresh Tools**.

This command refreshes the toolbar region of the window and the context menu for the **System Tray** icon by processing all the program shortcuts in the **Tools** folder.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Customizing the Toolbar Utility](#)

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Starting Applications with the Toolbar Utility](#)

Starting Applications with the Toolbar Utility

The **Toolbar Utility** is started automatically by the clients and is minimized as an icon in the **System Tray**. Once the **Toolbar Utility** is started, you can use to launch applications in one of two ways:

- ◆ Using the toolbar in the **Toolbar Utility** window.
- ◆ Using the context menu on the **Toolbar Utility** icon in the **System Tray**

To launch applications from the Toolbar Utility toolbar

- 1 Open the **Toolbar Utility** by doing one of the following:
 - ◆ Double-click the **Toolbar Utility** icon on the **System Tray**.
 - ◆ Right-click the **Toolbar Utility** icon on the **System Tray** and choose **Open**.

Note: When no portion of the **Toolbar Utility** user interface is visible, you can display its expanded window by starting a second instance of it. To do so, choose **Start ▶ Programs ▶ StarTeam ▶ StarTeam Toolbar**.

- 2 On the **Toolbar Utility** toolbar, click the toolbar button for the application you want to open.

To launch an application from the Toolbar Utility icon in the System Tray

- 1 Right-click the **Toolbar Utility** icon in the **System Tray**.
- 2 Choose the application you wish to open from the context menu.

Related Concepts

[Toolbar Utility](#)

Related Procedures

[Opening and Closing the Toolbar Utility](#)

[Logging On to Servers with the Toolbar Utility](#)

[Customizing the Toolbar Utility](#)

Managing Projects

This section contains procedures related to managing projects.

In This Section

[Assigning Access Rights to Projects](#)

Describes how to assign access rights to a project.

[Changing Project Names or Descriptions](#)

Describes how to change a project name or description.

[Configuring Projects to use APEs](#)

Describes how to configure a project to use Alternate Property Editors.

[Configuring Server Configurations to use Enhanced Process Links](#)

Describes how configure a server configuration to use enhanced process links by default for all projects.

[Creating Projects](#)

Describes the steps for creating a project.

[Creating Reports](#)

Describes how to create reports.

[Deleting Projects](#)

Describes how to delete a project.

[Demoting View Labels](#)

Describes how to demote a view label.

[Displaying Location References](#)

Describes how to view folder or item location references.

[Enabling Keyword Expansion](#)

Describes how to enable keyword expansion

[Enabling the use of Enhanced Process Links](#)

Describes how to enable the use of enhanced process links in the client for a project.

[Establishing Process Rules for Projects](#)

Describes how to require that files added to or checked in to a project be linked to a process item.

[Opening Existing Projects](#)

Describes how to open an existing project.

[Opening Projects with Shortcuts](#)

Describes how to use shortcuts to open projects.

[Promoting View Labels](#)

Describes how to promote a view label from one promotion state to the next.

[Requiring Exclusive Locks for Check-ins](#)

Describes how to force users to lock files before checking them in.

[Requiring Revision Comments](#)

Describes how to require revision comments when checking files in.

[Saving Projects as Shortcuts](#)

Describes how to create a shortcut to a project view.

[Setting Active Process Items](#)

Describes how to set an active process item to make it the default process item for future file additions or check-in operations.

[Setting Folder Component Access Rights](#)

Describes how to set folder access rights.

[Switching Views](#)

Describe how to switch to another project view.

[Viewing Connection Properties](#)

Describes how to view the server connection properties.

[Viewing or Modifying Project Properties](#)

Describes how to view or change project properties.

Assigning Access Rights to Projects

Project-level settings define access rights for all views, child folders, and items in the project. A user will not be able to open a project if you deny any of the following to a user or to all the groups to which the user belongs:

- ◆ The ability to see the project.
- ◆ The ability to see the initial (or root) view of the project.
- ◆ The ability to see the root folder of the initial view of the project.

Note: It is critically important to define a complete set of project-level access rights for all groups with view rights to a project. By default, groups with view rights have complete access to categories that have no rights defined. Therefore, it is best to specify access rights across all project rights for all groups.

Any time you create one grant record for a node, you should also create a grant record for that node for every group that will access the project at this level. It is also a good idea to include Administrators on each node. Then if permissions are ignored, an administrator can still change access rights, and so on.

To set project access rights

- 1 Open the project and select **Project** ► **Access Rights** from the StarTeam client menu. This action displays the **Project Access Rights** dialog box.
- 2 Select a node.
It is best to start with the **Project** node and work down.
- 3 Click **Add** to add a user or group. The **Assign Access Rights To** dialog box opens.
- 4 Select a user or group. Users are listed by their user names and groups are listed by their position in the group hierarchy, except for the All Users group.
- 5 Select the **Grant** option button.

Note: Never select the **Deny** option button unless you are creating an exception.

- 6 Click **OK** to return to the **Project Access Rights** dialog box.
- 7 Select and/or clear the appropriate check boxes for the user or group.
- 8 Repeat steps 3-7 as required for additional nodes, then click **OK** when you are done.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Creating Projects](#)

[Adding Files to Projects](#)

[Adding Server Configurations](#)

Changing Project Names or Descriptions

If you have the appropriate access rights, you can use the Project Properties dialog to review or change the project name and description.

To change a project name or description

- 1 Select **Project** ► **Properties** from the menu. The **Project Properties** dialog opens.
- 2 Select the **Name** tab.
- 3 Type a new project name in the **Name** text box.
- 4 Type a new project description in the **Description** text box.
- 5 Click **OK**.

Related Concepts

[Overview of Projects](#)

[Creating Projects](#)

Related Procedures

[Adding Files to Projects](#)

[Adding Server Configurations](#)

Configuring Projects to use APEs

If your company has licensed StarTeam Enterprise Advantage, you can use alternate property editors (APEs). APEs are Java forms created specifically for your company to support a corporate process. Workflow processes are created for use with the forms. APEs can be created for the File, Change Request, Requirement, Task, and Topic components.

APEs use the StarTeam SDK to access a StarTeam server configuration. APEs can be customized because they are implemented in standard programming languages. Sample property editors for several StarTeam components are included with StarTeam Enterprise Advantage..

To configure a project to use APEs with items

- 1 Select **Project** ► **Properties** from the menu. The **Project Properties** dialog appears.
- 2 Select the **Editors** tab.
- 3 Check **Use Alternate Property Editors for [item]** and browse for the correct APE.
- 4 Generally, you enter the word **Locator** followed by the name of the APE. **Locator** is a program on each client workstation that distributes code and XML updates to client workstations.

When **Locator** is specified and StarTeam requests the APE, **Locator** looks for the StarTeam Extensions project managed by the server configuration. If the project exists and contains an APE with the specified name, Locator copies the tip revision of that APE and its related files to the client workstation (if they are not already there). Then the APE will be used instead of the StarTeam standard dialog for that type of item.

Note: Take care when setting the APE for a project in production because the new setting takes effect immediately. It is important to test any changes in an APE prior to making this editor available to a wide audience.

Related Concepts

[Overview of Projects](#)
[Creating Projects](#)

Related Procedures

[Adding Files to Projects](#)
[Adding Server Configurations](#)

Configuring Server Configurations to use Enhanced Process Links

A server administrator can specify that the default server behavior allows the use of enhanced process links for all projects on the server configuration, and/or for all new projects created on a server configuration.

In the standard linking model, if a given item is specified as the reason for a change, then process links are created directly from that process item to each changed file or folder. In the enhanced linking model, the process item (the item specified as the reason for making a given set of changes) is distinguished from the task that represents the act of making the associated changes in a particular view. Changes are linked to the process item indirectly, through a process task.

Note: Borland recommends that you either specify both the **Enable enhanced links for all projects** and the **Enable enhanced links for new projects** options on the **General** tab in the **Configure Server** dialog box, or leave both options deactivated for the specified server configuration. You can then control the usage of enhanced process links on a per-project-basis in the **Process Rules** tab of Cross-Platform Client's **Project Properties** dialog box. For example, if you activate both options for the server configuration, then you could turn off this option in the **Process Tab** of the Cross-Platform Client's **Project Properties** dialog box by clearing the option to **Enable enhanced process links**. Once you start controlling these options per project in the **Project Properties** dialog box, the **Enable enhanced links for all projects** check box is no longer activated for the server configuration.

To specify that all projects on a server configuration be enabled to use enhanced process links

- 1 Open the Server Administration tool.
- 2 Log on to the server.
- 3 Choose **Tools** ► **Administration** ► **Configure Server** and select the **General** tab.
- 4 Do one or both of the following:
 - ◆ Check **Enable enhanced links for all projects**. This will allow users to enable and disable enhanced process links for all projects on that server configuration.
 - ◆ Check **Enable enhanced links for new projects**. This will allow users to enable and disable enhanced process links for new projects created on that server configuration.

Note: On server configurations created prior to StarTeam Server 2008, only the **Enable enhanced process links by default for all projects** will be visible since the enhanced-model new projects is not an available option for older server configurations.

Related Concepts

[Overview of Projects](#)

[Process Items and Process Links](#)

Related Procedures

[Enabling the use of Enhanced Process Links](#)

Creating Projects

To create a project

- 1 Start the StarTeam client.
- 2 Choose **Project** ► **New** to open the **New Project Wizard**.
- 3 Select a server configuration for the project from the server list.

Note: If the server you want is not in the **Server/Project Tree**, click **Add Server** and add it using the **Add Server** dialog box.

- 4 Click **Next** to continue.
- 5 Type the name and description for the new project in the **Project Name** and **Project Description** fields, and click **Next**.
- 6 Browse to the folder on your computer that will be the default working folder for the project root folder.
If the working folder does not exist, type the folder path and name and the **New Project Wizard** will create it.

Note: The default working folder must point to a location that is physically discrete for each user, such as a drive on your local computer or a personal directory on a shared file server.

- 7 Click **Next**.
- 8 Optionally, if the working folder has child folders, select any child folders you do not want added to the project and click **Exclude**.

Tip: Click **Reset** to include the previously excluded folders.

- 9 Click **Finish** to complete and open the project.

Note: After you create a project, the StarTeam client window displays a hierarchical **Folder Tree** of folders in the current view of the project. You can add other folders, if desired.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Files to Projects](#)

[Adding Server Configurations](#)

Creating Reports

All reports that you create in the application show all or some portion of the data displayed in the upper pane. All reports are generated in .html format.

Note: You can also create reports using StarTeam Datamart. See the “StarTeam Datamart User Guide.pdf” for more information.

To create reports

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Choose **Reports** from the component or context menu. The **Reports** dialog displays the **Available Reports** list box.
- 4 From the **Available Reports** list box, choose the type of the report you want to generate.
- 5 Do one of the following:
 - ◆ To include only the items selected on the upper pane, select the **Current Selection** option button.
 - ◆ To include all items displayed in the upper pane, select the **Select All** option button.
- 6 Type or browse to the path and report filename in the **Output file name** text box.
Be sure to use `.htm` or `.html` as the file extension.

Note: By default, the default report filename uses the convention
`<STReport><date><alphanumeric code>.html` (e.g.
`STReport2006-07-24T22-03-59Z.html`).
- 7 Type a name for your report in the **Report title** text box.
- 8 Click **Generate** to view the report on screen. Your Web browser opens and displays your report. In addition, the report is saved in the location specified in step 6 above.

Related Concepts

[Reports](#)

[Data Export with Datamart](#)

Related Procedures

[Customizing Report Templates](#)

[Configuring the Report Output Path](#)

[Printing Reports](#)

Deleting Projects

To delete a project, you must have the delete privilege or access right. Be absolutely certain that you want to delete a project and its folders because you and other members of your team will no longer be able to access any item in the project.

After a project is deleted, it will not be visible in the **Open Project** or **Select View** dialogs. If other users are connected to it at the time it is deleted, they will receive a message the next time they initiate a project or view command.

Deleting a project does remove any data from the server database. However, items that are not shared will no longer be accessible.

To delete a project from the server

- 1 Select the root project folder.
- 2 Select **Project** ► **Delete**. A message displays, asking you to confirm the deletion.
- 3 Click **Yes** to delete the project.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Files to Projects](#)

[Adding Server Configurations](#)

[Supporting Cross-Project Activities](#)

Demoting View Labels

Sometimes a labeled set of files is promoted prematurely and must be demoted. For example, if a specific build is promoted to the Beta state, but contains serious flaws, it should probably be returned to the prior promotion state. You can only demote view labels by editing the promotion state.

To demote a view label to a previous state

- 1 Choose **View** ► **Promotion** to display the **Promotion** dialog box.
- 2 Click Edit to open the **Promotion State** dialog box.
- 3 Select a different view label from the **View Label** drop-down list.
- 4 Click **OK**.

Related Concepts

[Overview of Views](#)

[Labels](#)

[Folders](#)

[Understanding View Types](#)

[Proper Use of Views](#)

[Understanding Branching](#)

Related Procedures

[Rolling Back the Current View Configuration](#)

[Promoting View Labels](#)

[Managing Views](#)

[Creating and Configuring Views](#)

[Creating View Labels](#)

[Copying View Labels](#)

[Creating Revision Labels](#)

[Copying Revision Labels](#)

[Reviewing and Moving Labels](#)

[Freezing or Unfreezing Labels](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Displaying Location References

Because of manual sharing and because views are children of other views, a folder or item can be associated with more than one project, view, or parent folder (within the same server configuration). Each instance of the folder or item has a reference to its tip revision.

To view folder references, you open a separate dialog; to view item references, you use the project view window and the **Reference** tab on the lower pane.

To view references, do one of the following

- 1 View folder references
- 2 View references for past revisions of a folder
- 3 View item references
- 4 View references for past revisions of an item

To view folder references

- 1 Select the folder in the folder hierarchy tree.
- 2 Right-click the selected folder and choose **Advanced** ► **References**.

The **Folder References** dialog box opens and displays a tree that indicates which project views reference this folder, and their relationship to each other.

View references for past revisions of a folder

- 1 Right-click the folder in the folder hierarchy tree and choose **Properties**.
- 2 Click the **History** tab.

Note: There is no **History** tab if you do not have the access rights that allow you to see the folder history.

- 3 Select the revision in the **History** list.
- 4 Right-click the selected revision and choose **References**.

An appropriate **References** dialog box appears.

To view item references

- 1 Select an item in the upper pane.
- 2 Click the **Reference** tab beneath the lower pane.

Data similar to that in the **Folder References** display in the lower pane. The **Reference** pane has no context menu.

View references for past revisions of an item

- 1 Select an item in the upper pane.
- 2 Click the **History** tab in the lower pane.

- 3 Select the revision in the **History** list.
- 4 Right-click the selected revision and choose **References** from the **History** context menu.
An appropriate **References** dialog box appears.

Related Concepts

[References to Folders and Items](#)

Related Procedures

[Managing Projects](#)

[Working with Folders and Items](#)

Enabling Keyword Expansion

By enabling keyword expansion for a project, you can embed keywords within text files. These keywords are automatically expanded during file check-outs, to provide file and revision information within the file. You should use only one keyword per line.

Keyword expansion should work for all "ASCII-based" encodings, which includes UTF-8, Cp1252, and so on. It does not include the various UTF-16 encodings. StarTeam currently treats UTF-16 encodings as binary and does not attempt to perform either EOL or keyword expansion on them. The Cross-Platform Client allows you to select a file encoding as you check out a file.

Warning: Never use a keyword in a revision comment, as it will be expanded during the keyword expansion process.

To enable keyword expansion

- 1 Select **Project** ► **Properties** from the main menu.
The **Project Properties** dialog box opens.
- 2 Select the **Options** tab.
- 3 Check the **Keyword Expansion** check box to enable keyword expansion and use keywords in your text files.
This check box applies to files added or checked in from StarTeam or from StarTeam integrations with third-party applications.
- 4 In **Expand Keywords for These File Extensions** text box, type the file extensions (for example, `.bat`, `.cpp`) for which you want to use keywords.
You can use a space, comma, or semicolon as keyword delimiters. The file extensions list can contain a maximum of 254 characters. If you leave this text box blank, no keywords will be expanded.

Related Concepts

[Personal Options](#)

Related Reference

[Table of StarTeam Keywords](#)

Enabling the use of Enhanced Process Links

In the Cross-platform Client, each user can specify that a project uses the enhanced process link model instead of the standard linking model. In the standard linking model, if a given item is specified as the reason for a change, then process links are created directly from that process item to each changed file and folder.

In the enhanced linking model, the process item (the item specified as the reason for making a given set of changes) is distinguished from the process task that is automatically created to represent the associated changes in a particular view. Changed files and folders are linked to the process item indirectly, through a process task.

Note: Borland recommends that you either specify both the **Enable enhanced links for all projects** and the **Enable enhanced links for new projects** options on the **General** tab in the **Configure Server** dialog box, or leave both options deactivated for the specified server configuration. You can then control the usage of enhanced process links on a per-project-basis in the **Process Rules** tab of Cross-Platform Client's **Project Properties** dialog box. For example, if you activate both options for the server configuration, then you could turn off this option in the **Process Tab** of the Cross-Platform Client's **Project Properties** dialog box by clearing the option to **Enable enhanced process links**. Once you start controlling these options per project in the **Project Properties** dialog box, the **Enable enhanced links for all projects** check box is no longer activated for the server configuration.

To enable the use of enhanced process links for a project in the client

- 1 Choose **Project** ► **Properties** to open the **Project Properties** dialog box.
- 2 Select the **Process Rules** tab.
- 3 Check **Enable Enhanced Process Links**, then click **OK**.

Related Concepts

[Overview of Projects](#)

[Process Items and Process Links](#)

Related Procedures

[Creating Projects](#)

[Configuring Server Configurations to use Enhanced Process Links](#)

Establishing Process Rules for Projects

Establishing a system of process rules allows you to:

- ◆ Require that process items are used every time files are added or checked into the project.
- ◆ Stipulate that only certain types of items with specific statuses can be used as process items in the project.
- ◆ Enable the use of enhanced process links for the project.

Note: To set process rules, you must have the access rights required to change project properties. Usually, only team leaders and administrators have these rights. You must also verify that project users have the rights to see and modify items in the project view, to create and modify links on files and process items, and to create tasks and link to tasks if using the enhanced model..

To require use of process items at check-in, and specify which types to use

- 1 Choose **Project** ► **Properties** and click the **Process Rules** tab in the **Project Properties** dialog box.
- 2 Check **Require Selection Of Process Items When Files Are Added Or Checked In**.
- 3 Check which types of items you want to allow for use as process items. (See below.)

To permit the use of change requests as process items

- 1 Check **Permit Selection Of Change Requests As Process Items**.
- 2 Do one of the following:
 - ◆ Check **Open** to use only change requests with **Open** status.
 - ◆ Check **In Progress** to use only change requests with **In Progress** status.
 - ◆ Check both **Open** and **In Progress** to use only change requests with either of these statuses.
 - ◆ Uncheck both **Open** and **In Progress** to allow any change request to be used as a process item, regardless of status.

To permit the use of requirements as process items

- 1 Check **Permit Selection Of Requirements As Process Items**.
- 2 Do one of the following:
 - ◆ Check **Approved** to use only requirements with **Approved** status.
 - ◆ Uncheck **Approved** to allow any requirement to be used as a process item, regardless of status.

To permit the use of tasks as process items

- 1 Check **Permit Selection Of Tasks As Process Items**.
- 2 Do one of the following:
 - ◆ Check **Ready To Start** to use only tasks with **Ready To Start** status.

- ◆ Check **In Progress** to use only tasks with **In Progress** status.
- ◆ Check both **Ready To Start** and **In Progress** to use only tasks with either of these statuses.
- ◆ Uncheck both **Ready To Start** and **In Progress** to allow any tasks to be used as a process item, regardless of status.

To enable enhanced process links for the project

- 1 Choose **Project** ► **Properties** and click the **Process Rules** tab in the **Project Properties** dialog box.
- 2 Check or uncheck **Enable Enhanced Process Links** to specify whether to use the standard linking model or the enhanced linking model.

In the enhanced model, the process item (that is, the item specified as the reason for making a given set of changes) is linked to a process the task, which is automatically created to represent the associated changes in a particular view. Changes are linked to the process item *indirectly*, through a process task.

Note: Some StarTeam integrations do not recognize process rules and will ignore them.

Related Concepts

[Process Items Overview](#)

[Process Items and Process Links](#)

Related Procedures

[Assigning Access Rights to Projects](#)

Related Reference

[Access Rights and Privileges](#)

Opening Existing Projects

Before you can access an existing project view, you must have access to the appropriate server, be logged on, and open the project.

To open an existing project

- 1 Do one of the following:
 - ◆ Click **Open Project** on the **Project** toolbar.
 - ◆ Select **Project** ► **Open** from the menu. This action displays the **Open Project** wizard.
- 2 Click the plus sign in front of the server name or double-click the name of the server configuration on which the project is located.
 - ◆ If you have already logged on to this server configuration, skip to the next step.
 - ◆ If you are not logged on, the **Log On to [server configuration] [project name]** dialog opens.
- 3 Type your **User Name** and **Password**. You must have the necessary access rights to continue. Passwords are case-sensitive and may have length restrictions. See your StarTeam administrator for details.
- 4 Once you are logged in, the **Open Project** wizard displays a list of projects for the selected server configuration. Do one of the following:
 - ◆ Select a project name, then click **Finish** to open your project.
 - ◆ Double-click the project name to select a specific view of that project.
 - ◆ Select the project name, then click **Next** to select a specific view of that project.
- 5 If the **Select View** dialog opens from the **Open Project** wizard, select a name from the **View** list and click **Finish** or simply double-click the view name to open your project in that view.

Note: If the **View** icon is greyed-out, you do not have access to that view.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Files to Projects](#)

[Adding Server Configurations](#)

Opening Projects with Shortcuts

If you will be accessing a specific StarTeam project view frequently, you may want to create a shortcut for it on your desktop. Double-clicking the shortcut starts StarTeam and opens the view associated with the shortcut.

To open your project view with a shortcut

- 1 Select **Project Open Shortcut** from the **Project** menu. The **Open** dialog box opens.
- 2 Select the shortcut name and click **Open**.

Note: The View configuration is also saved as part of the shortcut.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Saving Projects as Shortcuts](#)

[Adding Files to Projects](#)

[Saving Projects as Shortcuts](#)

Promoting View Labels

You can promote a view label from one promotion state to the next if you have the appropriate access rights.

To promote a view label to the next promotion state

- 1 Choose **View** ► **Promotion** to open the **Promotion** dialog box which displays any states currently created for the view.
The states are displayed from the final state down to the initial state.
- 2 Select the promotion state currently associated with the view label that you want to promote.
- 3 Click **Promote**. The **Promote View Label** dialog box indicates that the view label is now associated with the next state (the state immediately above the selected state in the **Promotion** dialog box).
- 4 Verify that the information is what you were expecting to see, then click **OK**.

The selected view label now applies to two promotion states: the one to which it was promoted and the one you originally selected. Usually, your next action is to associate a new view label with the original state.

Related Concepts

[Overview of Views](#)
[Labels](#)
[Folders](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Understanding Branching](#)

Related Procedures

[Rolling Back the Current View Configuration](#)
[Demoting View Labels](#)
[Managing Views](#)
[Creating and Configuring Views](#)
[Creating View Labels](#)
[Copying View Labels](#)
[Creating Revision Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Demoting View Labels](#)
[Deleting Labels](#)
[Detaching Labels from Items](#)

Requiring Exclusive Locks for Check-ins

With this application, you cannot force users to lock files before they make changes. However, if you have the required privileges, you can require all users to conform with the policies and processes of your company by exclusively locking files before they check them back into the application.

Although this requirement helps users to avoid merge situations, they still must:

- ◆ Notice whether a file is already exclusively locked by another user before they check it out to work on it.
- ◆ Lock each file before making changes, to alert other users to their intentions.
- ◆ Be sure that the status for each working file is **Current** to avoid changing older revisions of the files. If a file status is not **Current**, the file must be checked out before any changes are made.

To force users to lock files before checking them in

- 1 Select **Project** ► **Properties** from the menu bar.
The **Project Properties** dialog opens.
- 2 Select the **Options** tab.
- 3 Check **Require Exclusive Lock When Files are Checked In**.

When this option is selected, only a person who has exclusively locked a file can check it in.

Note: If developers are using an application integration for a development environment, such as StarTeam Visual Studio Integration, they cannot check in files from that environment if both the **Require Exclusive Lock When Files are Checked In** check box in the **Project Properties** dialog box, and the **Use Non-exclusive Locks in Integrations** check box on the **Personal Options** dialog box (**File** tab) are selected. In this situation, uncheck **Use Non-exclusive Locks in Integrations** to check files in.

Related Concepts

[Overview of Projects](#)
[Process Items Overview](#)

Related Procedures

[Creating Projects](#)
[Viewing or Modifying Project Properties](#)
[Changing Project Names or Descriptions](#)
[Requiring Revision Comments](#)
[Marking Unlocked Files Read-only](#)

Requiring Revision Comments

When users check in files by using **File** ► **Check In** or selecting **Check In** from the context menu, the **Check In** dialog displays. By default, this dialog allows them, but does not require them, to type a comment about the operation.

If users check in files by clicking either of the **Check In** buttons on the toolbar, this dialog box does not open, so they cannot type a revision comment.

Administrators can force users to supply a check-in reason, however, by adjusting the properties for the project. This requirement will apply whether the users perform the check-in with the **File** menu, context menu, or toolbar.

To force users to supply a revision check-in comment or reason

- 1 Select **Project** ► **Properties** from the menu bar.
The **Project Properties** dialog box opens.
- 2 Select the **Options** tab.
- 3 Check **Require Revision Comment When Files are Checked In**.

Note: This check box applies only to the application, not to integrations.

- 4 Click **OK**.

From this time on, the **Check-in** dialog box will always open when users check in files, and they will need to type text in the **Reason for Check-in** text box before completing the operation.

Related Concepts

[Overview of Projects](#)
[Process Items Overview](#)

Related Procedures

[Creating Projects](#)
[Viewing or Modifying Project Properties](#)
[Changing Project Names or Descriptions](#)
[Requiring Exclusive Locks for Check-ins](#)

Related Reference

[Marking Unlocked Files Read-only](#)

Saving Projects as Shortcuts

If you will be accessing a specific StarTeam project view frequently, you may want to create a shortcut for it on your desktop. Double-clicking the shortcut starts StarTeam and opens the view associated with the shortcut.

To save a shortcut to your project view

- 1 Log onto StarTeam and open the project view window for which you wish to create the shortcut.
- 2 Select **Project Save Shortcut As** or click **Save Shortcut** on the toolbar. These actions open the **Save As** dialog.
- 3 Use the default name or type another name for the shortcut in the **File Name** text box.
Be sure to keep the `.stx` extension.
- 4 Select a location, usually your desktop, for storing the shortcut.
- 5 Click **Save**.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Files to Projects](#)

[Opening Projects with Shortcuts](#)

Setting Active Process Items

Setting an active process item is a convenient way of saving time when you know that you will be adding files or checking them in later. When you have an item selected on the upper pane, making it the active process item is a simple operation.

You can make a selected change request, task, or requirement the active process item for the current view, an open view on the same server, or a different view on the same server.

Note: You can only specify one active process item for each view. Setting a second active process item for the same view at one time clears the first one.

To set an active process item

- 1 Select a folder in the folder hierarchy.
- 2 Click the **Change Request**, **Requirement**, or **Task** tab and select the item you want to use as the active process item.
- 3 Do one of the following:
 - ◆ Right-click and choose **Set Active Process Item** ► **Current View** to choose the current view.
 - ◆ Right-click and choose **Set Active Process Item** ► **[view name]** to choose from the listed opened views on the server.
 - ◆ Right-click and choose **Set Active Process Item** ► **Select View** to open a dialog box and choose any other view on the server.

Note: You can also set the currently-selected item as a process item by using the **Use As Active Process Item** button on the toolbar.

The active process item you selected is used by default when you add files or check them in. However, you can override this default and select another appropriate item when adding or checking in files.

Tip: After you finish with a process item, you should right-click it and choose **Clear Active Process Item** so that it cannot be accidentally reused. That removes the information from the status bar and keeps the process item from reappearing in the **File Add** or **File Checkin** dialog boxes.

Related Concepts

[Process Items Overview](#)

[Active Process Items](#)

[Process Tasks and Enhanced Process Links](#)

Setting Folder Component Access Rights

To set access rights for the Folder component

- 1 Click the **Folder** tab in the **Item** pane.
- 2 Right-click the folder and choose **Advanced** ► **Component Access Rights**.

Note: You must have the administrative rights to change component access rights for a project.

- 3 In the **Folder Component Access Rights** dialog box, select a component from the **Category** list, then click **Add**.
- 4 In the **Assign Access Rights To** dialog box, select a group and/or user and **Grant** or **Deny** them access.
- 5 Click **OK** to return to the **Folder Component Access Rights** dialog box.
- 6 Select the group or user you just added to the **Users and Groups** list, then check the desired options in the **Rights** column, or click **Select All**.
- 7 Choose **Grant** or **Deny** to specify what action to take for the selected users and rights.
- 8 Continue this process until you have finished granting or denying access rights to the appropriate users and groups, then click **OK**.

Switching Views

When you open a project, you select a view of that project. Once a project is open, however, you can switch to another view. By default, the newly selected view always opens with the Current configuration, regardless of the configuration it had when you last exited it.

To switch to another project view

- 1 On the **View** menu or context menu, choose **Select View**. The **Select a View** dialog box opens, which shows the views hierarchically.
- 2 Check **Open in New Window** if you would like to keep your existing view open rather than changing the window to the new view. When you check this item, a new view window will open in the Cross-Platform Client window and the name of the new view will be added to the list of opened views on the **Window** menu.
- 3 Select a name from the **View** list and click **OK**, or simply double-click the view name to open your project in that view.

If you do not wish to use the default configuration, you can roll back the current view configuration.

Note: If you wish to open two different StarTeam view windows in the same project at the same time, select **Project ▶ Open** to display the additional view.

Related Concepts

[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)
[Copying View Labels](#)
[Deleting Views](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Viewing Connection Properties

Reviewing the connection properties lets you verify the server, address, end point (port), logged on user, and logon time.

To check connection properties

- 1 Select **Project** ► **Connection Properties** from the menu bar.
The **Connection Properties** dialog box appears displaying all the server connection property values.
- 2 Click **OK**.

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Server Configurations](#)

[Starting Applications with the Toolbar Utility](#)

[Logging Off](#)

Viewing or Modifying Project Properties

Each project can be configured with properties that allow you to specify the project name, enable keyword expansion, require revision comments on check-in, specify file locking behavior on check-in, define process rules and process items, and specify alternate property editors.

To view or modify project properties

- 1 Choose **Project** ► **Properties** to open the **Project Properties** dialog box.
- 2 Select a tab containing properties you want to view or change.
- 3 Make all desired changes, then click **OK**.

Related Concepts

[Overview of Projects](#)
[Process Items Overview](#)

Related Procedures

[Creating Projects](#)
[Changing Project Names or Descriptions](#)
[Requiring Revision Comments](#)
[Requiring Exclusive Locks for Check-ins](#)
[Marking Unlocked Files Read-only](#)

Managing Views

This section contains tasks related to managing views.

In This Section

[Changing a View's Default and Alternate Working Folders](#)

Describes how to change the Default and Alternate Working Folders for a view.

[Copying View Labels](#)

Describes how to create a view label based on an existing view label.

[Creating and Configuring Views](#)

Describes how to create new views.

[Creating View Labels](#)

Describes how to create a view label.

[Deleting Views](#)

Describes how to delete views.

[Modifying View Names or Descriptions](#)

Describes how to change the name and description for a view.

[Refreshing Views](#)

Describes various ways to refresh views in StarTeam.

[Reviewing or Modifying View Properties](#)

Describes how to view or modify view properties.

[Rolling Back the Current View Configuration](#)

Describes how to roll back the current view configuration.

Changing a View's Default and Alternate Working Folders

Make sure that everyone is logged off from the server and that the server is locked before you change the **Default Working Folder**. It is just as critical to perform these actions as it is when you change custom fields or do anything else that affects all users.

When you change the **Default Working Folder**, not only the path to the working folder but the path to each child folder in the view may be similarly modified—not just for you, but for everyone working with that view.

Warning: Do not change the **Default Working Folder** unless you are a project administrator. These default settings affect ALL users and incorrect settings cause other users to be unable to check out StarTeam files. The default settings should only be set to the name of the folder. If you want to use a different location for your working folder than the **Default Working Folder** path, specify an **Alternate Working Folder** path.

To change the working folder

- 1 Choose **View** ► **Properties** to open the **View Properties** dialog box.
- 2 Select the **Name** tab.
- 3 Do one of the following:
 - ◆ Select **Alternate** to create a different working folder for only yourself.
 - ◆ If you are a project administrator, select **Default** to specify the default repository path for all users.
- 4 Type the name of a new working folder or browse for a path to a working folder. If you browse for the path, it becomes an absolute path. This path can be edited, however, to enable you to work on a computer that uses a different letter for its hard drive.

Note: It is important that the **Default Working Folder** point to a location that is physically discrete for each user, such as a drive on that user's workstation or a personal directory on a shared file server.

Related Concepts

[Overview of Views](#)
[Understanding Default and Alternate Working Folders](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)

Copying View Labels

Occasionally, you may want to create a view label and attach it to the same item revisions as an existing view label, with a few additions or exceptions. The steps in this procedure explain how to create a view label based on an existing view label. For example, suppose builds are done only after a view has been rolled back to a label and that the build is given the same name as the label. If, in the last build, only one Help file was missing, you would probably change the existing label to include that one file and rebuild. However, if the previous build was already made available to users participating in a field test, using the same label could cause confusion. It would be better to create a new view label as a copy of the older label and then add the missing file to the new label.

Note: You cannot copy a view label unless it already exists in the view in which you are performing this operation. The view configuration must also be current.

To create a new view label and attach it to the same item revisions as the existing view label

- 1 Choose **View** ► **Labels** to open the **Labels** dialog box.
- 2 Click **New** to open the **View Label** dialog box.
- 3 Type a name and description for the label.
- 4 Select the **Labeled Configuration** option to attach the label to item revisions that have an existing label.
- 5 Optionally, uncheck **Use As Build Label** if you do not want this label to be a build label.

Note: By default all view labels are designated as build labels.

- 6 Click **OK**, then click **Close** to exit the **Labels** dialog box.
The new view label is now attached to the same revisions as the existing label.
- 7 Select the items in the upper pane for which the new label must differ.

Tip: You can also select all items with a specific label. Right-click in the upper pane, choose **Select** ► **By Label**. When you select the label, all the items attached to that label are automatically selected.

- 8 Detach the new label from items that you do not want to include.
- 9 Attach the new label to items formerly not included, and/or attach the new label to different revisions of items to which it is already attached.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Copying Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

Creating and Configuring Views

This topic provides the basic procedure for creating a new view based upon an existing view.

To create and configure a new view

- 1 Display the project view upon which the new view will be based.
- 2 Choose **View** ► **New** to open the **New View Wizard**.
- 3 Select one of the available options from the **View Type** list box.
 - ◆ **Branch All**: Will be based on a configuration of the currently open view. All items in the new view will be set to branch when they are modified. This branch behavior can be changed later for individual items in the new view so that changed items do not branch.
 - ◆ **Reference**: Allows users to read from and write to a subset of the parent view's current configuration. Any changes appear in both the reference view and its parent.
 - ◆ **Read-only Reference**: Allows users to read from a subset of the parent view. Unlike a read-write reference view, the contents of a read-only reference view may be current (floating) or configured to a point in the parent view's past by specifying a label, promotion state, or time.

For more details on view type options, including advanced view types, see "View Type Options and Settings" in the links below.

- 4 Type a **Name** and a **Description** for the view in the appropriate text boxes and click **Next**.
- 5 Select the **Root Folder** for the new view and click **Next**.

Note: The **New View Wizard** skips this step for a **Non-Derived** view.

- 6 Type or browse for the name of an appropriate **Default Working Folder**.

Warning: For a **Branch All** or a **Non-Derived** view, always use a working folder that is different from the one used by the parent view. Using the same working folder for the parent and child views can cause changes in one view to be overwritten when files are checked out from the other view. It can also result in incorrect or, at least, misleading file status indicators. For a **Reference** or **Read-Only Reference** view, you can use the same working folder as the parent view.

- 7 Click **Next** to display the **Select Types** page.
- 8 Select the item types to include in the new view. The new view will include items of the selected types from the parent view, unless the **Override default types** option has been selected in the **View Access Rights** dialog box.

Tip: To create a view with no shared items, use the **Branch All** view type and clear all the check boxes on the **Select Types** page.

- 9 Click **Next** to display the **Configuration** page.
If you are creating a **Non-derived** or **Reference** view, click **Finish**.

Note: It is not necessary to display the **Configuration** page for a **Non-Derived** view because no items from the parent view are included. It is also not necessary to display the **Configuration** page for a **Reference** view because the items have the same configuration as those in the parent view.

- 10 Select one of the available configuration options on the **Configuration** page.
- 11 Click **Finish**.

Related Concepts

- [Overview of Views](#)
- [Understanding Default and Alternate Working Folders](#)
- [Understanding View Types](#)
- [Proper Use of Views](#)
- [Understanding Branching](#)
- [Overview of Branching Options](#)
- [Branching Behavior of Items](#)

Related Procedures

- [Managing Views](#)
- [Changing a View's Default and Alternate Working Folders](#)
- [Copying View Labels](#)
- [Deleting Views](#)
- [Rolling Back the Current View Configuration](#)

Related Reference

- [View Type Options and Settings](#)
- [View Configuration Options](#)

Creating View Labels

View labels, usually used as build labels by default, can be extremely useful when you want to label every folder and item in a particular view.

To create a view label

- 1 Open the view to which you want to apply the label.
- 2 Choose **View ▸ Labels**.
The **Labels** dialog box opens with the **View** tab already selected. This tab lists existing view labels in reverse chronological order, based on the time when they were created.
- 3 Click **New** to create a new label and add its name to the list box.
The **View Label** dialog box opens.
- 4 Type a name and description for the label in the appropriate text boxes. The maximum label name length is 64 characters, and the description length is 254 characters.
- 5 Select one of the following options:
 - ◆ **Current Configuration**: This attaches the label to the tip revision of every item in this view's current configuration.
 - ◆ **Labeled Configuration**: This option attaches the label to the revisions of items to which the label you specify is currently attached. (Creating a view label based on another view label is equivalent to copying that view label.)
 - ◆ **Promotion State Configuration**: This option attaches the label to the revisions of items currently in the promotion state that you specify. (Actually, the label is attached to the revision that has the promotion state's current view label.)
 - ◆ **Configuration As Of**: This option attaches the label to the revision of every item that was a tip revision at the specified date and time.
- 6 Optionally, check **Use As Build Label** to update each change request that has **Next Build** as the setting for its **Addressed In Build** property. If this option is not selected, the view label will still be attached to change requests, but the setting of the **Addressed In Build** property will not change.
- 7 Optionally, to freeze the label so that the revisions attached to it cannot be changed, check **Freeze**.
- 8 Click **OK**.

Note: It is always important to synchronize the dates and times of the computers that run the StarTeam clients and the StarTeam server. However, if they are not synchronized and you select the current time as a label's configuration, the label may not be immediately visible.

Related Concepts

[Labels](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Creating Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

Deleting Views

Before deleting a project view, be absolutely certain that you wish to do so. After performing this operation, you will no longer be able to access any item in the view that is not shared with another project or view. Deleted views are also not visible in the **Select View** dialog box, although deleting a view does not remove any data from the server database.

If other users are connected to the project view when it is deleted, they will receive a **Deleted** message the next time they initiate a view command.

Note: A view cannot be deleted if it has derived child views.

To delete a view

- 1 Choose **View** ► **Delete**. The message box asks you to confirm the deletion.
- 2 Click **Yes** to confirm. A confirmation dialog box asks you to type the name of the project.
- 3 Type the project name, which is case-sensitive, in the **View Name** text box.
- 4 Click **OK**. The window for the deleted view closes.

Related Concepts

[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Creating and Configuring Views](#)
[Managing Views](#)
[Creating and Configuring Views](#)
[Copying View Labels](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Modifying View Names or Descriptions

Root views initially are assigned the same name as the project. If you have the appropriate privileges, however, you can change the name and/or description of the root view or its children.

To modify a view name and description

- 1 Choose **View** ► **Properties** to open the **View Properties** dialog box.
- 2 On the **Name** page, type a new name in the **Name** text box.
- 3 Type a new description in the **Description** text box.
- 4 Click **OK** to exit.

Related Concepts

[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)
[Copying View Labels](#)
[Deleting Views](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Refreshing Views

You can refresh data in the StarTeam project view window in several different ways, depending upon what you wish to update:

To refresh a view

1 From the view window, do one of the following:

- ◆ Press **F5** to refresh the upper right pane for the current item list, such as the File list or Change Request list.
- ◆ Press **CTRL+F5** to refresh the upper pane and simultaneously collapse all open groups.
- ◆ Press **SHIFT + F5** to refresh the entire view (all item lists in all tabs as well as the folder hierarchy).

Tip: You can turn on **Auto Refresh** in the **Personal Options** dialog box on either the **Workspace** tab or the **StarTeamMPX** tab. **Auto Refresh** is designed to perform even if the window is minimized.

Related Concepts

[Overview of Views](#)

[Understanding View Types](#)

[Proper Use of Views](#)

[Overview of Branching Options](#)

[Branching Behavior of Items](#)

Related Procedures

[Creating and Configuring Views](#)

[Rolling Back the Current View Configuration](#)

[Managing Views](#)

[Copying View Labels](#)

[Deleting Views](#)

Related Reference

[View Type Options and Settings](#)

Reviewing or Modifying View Properties

Sometimes you may want to look at the values and properties originally used to create a view. Reviewing this information may help you understand the behavior of changes within the view or the views that have been derived from it. Also, if you have access rights to do so, you may be able to modify view properties as well.

To review or modify view properties

- 1 Choose **View** ► **Properties** to open the **View Properties** dialog box.
- 2 Select the **Name** tab to see or change the following:

- ◆ Name and description of the view
- ◆ Who created the view and when it was created (read-only)
- ◆ Whether the items are set to branch on change
- ◆ Whether a central or a per folder repository is being used
- ◆ Working folder path

Note: Your access rights determine which items you can change.

- 3 Select the **Hierarchy** tab to review the list of views for this project and their relationship to one another.
- 4 Select the **Type** tab to see:
 - ◆ View type
 - ◆ Whether the view is a root, branching, non-derived, or a reference view
 - ◆ For branching views, whether the original default was Branch All or Branch None
 - ◆ Parent view on which this view was based
 - ◆ Parent configuration used to create this view
- 5 Click **OK** or **Cancel** to exit.

Related Concepts

[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)
[Copying View Labels](#)
[Deleting Views](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Rolling Back the Current View Configuration

By default, a view has a current configuration – that is, it displays the latest revisions of the items in the project. However, you can roll back a view to a past state based on a label, promotion state, or a point in time. Note that rolling back a view in this way configures it for the current user only.

When you roll back a view, this action prevents it from changing, until you select **Current Configuration** or close the project, which automatically changes the view to Current. You cannot check in files, update change requests, and so on in a rolled-back view because you cannot change the past.

When you configure a view, you can base it on a promotion state. Whether the state is assigned to **Current** or to a specific view label, any view configured to a promotion state is read-only. This read-only status can create problems for a user who needs to, for example, both look at files as they existed earlier and create new change requests. Because CRs can only be added to a current configuration, the user may need to have two windows open for the view: one configured to a point in the past and one set to the current configuration.

To roll back a current view

- 1 Choose **View** ► **Select Configuration** to open the **Select a View Configuration** dialog box.
- 2 Select a view configuration option:
 - ◆ **Labeled Configuration:** This option limits the view to items with a specified view label. It is disabled if the view has no labels. If the view is a branching view, the label must have been created in this view. Existing labels list in reverse chronological order, based on the time for which they were created. Reference views inherit labels from their parent views.
 - ◆ **Promotion State Configuration:** This option rolls back the view to a specific promotion state. The view will display only those items that have the specified promotion state.
 - ◆ **Configuration As Of:** This option rolls back the view to a specific point in time. The view will display all items as they existed just prior to the specified date and time. This option defaults to the current date and time, but you can select a date and time in the past.
- 3 Click **OK**.

To return to the current configuration

- 1 Choose **View** ► **Select Configuration** to open the **Select a View Configuration** dialog box.
- 2 Select **Current Configuration** and click **OK**.

To base a view configuration on a promotion state

- 1 Choose **View** ► **Select Configuration** to open the **Select A View Configuration** dialog box.
- 2 Select **Promotion State Configuration**.
- 3 Select a state from the drop-down list box.

Configuring a view to a promotion state maintains the dynamic nature of the promotion state. If the promotion state view label changes, the view is configured to the new view label on the next refresh. The promotion state name appears on the status bar. If you exit the view and return, you return to the current configuration
- 4 Click **OK**.

Related Concepts

[Overview of Views](#)

[Understanding View Types](#)

[Proper Use of Views](#)

[Overview of Branching Options](#)

[Branching Behavior of Items](#)

Related Procedures

[Creating and Configuring Views](#)

[Managing Views](#)

[Copying View Labels](#)

[Deleting Views](#)

Related Reference

[View Type Options and Settings](#)

Branching Operations

This section contains information related to branching..

In This Section

[Configuring a View to Display Non-Branched Files](#)

Describes a quick way to display non-branched files and rebase them to a newer configuration from the parent view.

[Configuring the Branching Behavior of Shared Items](#)

Describes how to configure the branching behavior of shared items.

[Creating a Branching View](#)

Describes how to branch a view

[Reviewing or Changing Branching Behavior](#)

Describes how to review or change branching settings for files, folders, and change requests.

Configuring a View to Display Non-Branched Files

If you use a branch-all variant view with item configurations set to a specific timestamp, you may periodically want to catch up the parent view with changes in the branched (child) view by performing a Rebase operation. While you can do this with View Compare/Merge, when a branch-all variant view is used as an *activity* view, many files will not have been modified and hence will not have been branched. In this case, the non-branched files can be more quickly rebased with the main view by simply altering their configuration timestamp.

This procedure is a quick way to pull out all files that have not branched yet, and create a new view label to see them separately. This allows you to rebase the view on a controlled basis. The branched files, of course, have to be merged separately.

To display non-branched files, and rebase them to a new configuration from the parent view

- 1 Right-click the column heading again and choose **Sort and Group**.
- 2 Click **Show Advanced Fields** in the **Sort and Group** dialog box.
- 3 Change the sorting/grouping order so that the display groups by **Branch State**, followed by whatever else you normally like to see (such as sort by **Status**, then **Name**, and so on.)
- 4 Click **OK**.
The non-branched files will be displayed in the upper pane.
- 5 Right-click the non-branched files and choose **Advanced ▶ Behavior**.
- 6 Click the **Configuration** tab of the **Folder Behavior** dialog box.
- 7 Change the **Configuration Date** to a new timestamp, for example, the latest view label and click **OK**.

Tip: It is helpful to set up custom filters with the appropriate sort and group behaviors to make this easy.

Related Concepts

[Overview of Branching Options](#)
[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Understanding Branching](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)
[Changing a View's Default and Alternate Working Folders](#)
[Copying View Labels](#)
[Deleting Views](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Configuring the Branching Behavior of Shared Items

Folders and items can be shared from one view into another if both views belong to the same server configuration. They can also be shared from one folder to another within the same view. When you share folders and items that can branch, they acquire branching behavior in the new view. Requirements, tasks, and topics do not have branching behavior.

An item's behavior determines whether the item branches on change. A shared item's initial behavior in the new location depends upon the setting of the "Set items shared into view to branch on change" property when the item was shared. This view property appears on the View Properties dialog in the root view and in branching views, but it does not appear in reference views. In reference views, the behavior of the shared folders and items that can branch depends on this property's setting in the parent view of the reference view.

To configure the branching behavior of items

- 1 Choose **View** ► **Properties** to open the **View Properties** dialog box.
- 2 Click the **Name** tab.
- 3 Check or clear the **Set Items Shared Into View To Branch on Change** checkbox.

Note: Clearing the **Set Items Shared Into View To Branch on Change** checkbox is not recommended and will cause a warning message to be displayed upon saving.

- 4 Click **OK** or **Apply**.

After items have been shared into a view, you can change their behavior on an item-by-item basis, but additional changes to the **Set Items Shared Into View To Branch on Change** property do not change the behavior of the items.

Related Concepts

[Overview of Branching Options](#)
[Overview of Views](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Understanding Branching](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)
[Changing a View's Default and Alternate Working Folders](#)
[Copying View Labels](#)
[Deleting Views](#)
[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Creating a Branching View

A branching view is a new view derived from an existing view. When a branching view is created, StarTeam shares items from the parent view into the child view. The child view knows that the items came from the parent, but has no idea where the parent got them. For example, if a folder was shared to the parent view from another project, the child view does not have a similar share to the other project. If that is needed, the folder in the child view should be deleted and the folder should be shared to the child view from the other project (just as it was once shared to the parent view).

Similarly, if a folder was shared from the parent view to another project--or to another location within the parent view--the corresponding folder in the child view would not have that same relationship with the other project or view folder.

Note: Only folders, files, and change requests can branch. Requirements, tasks and topics never branch.

To branch a view

- 1 Display the project view upon which the new view will be based.
- 2 Choose **View New** to open the **New View** wizard.
- 3 Select **Branch All** from the **View Type** list box.
- 4 Type a **Name** and a **Description** for the view in the appropriate text boxes and click **Next**.
- 5 Select the **Root Folder** for the new view and click **Next**.
- 6 Type or browse for the name of an appropriate **Default Working Folder**.

Warning: For a **Branch All** view, always use a working folder that is different from the one used by the parent view. Using the same working folder for the parent and child views can cause changes in one view to be overwritten when files are checked out from the other view. It can also result in incorrect or, at least, misleading file status indicators.

- 7 Click **Next** to display the **Configuration** page.
- 8 Select one of the available configuration options on the **Configuration** page.
- 9 Click **Finish**.

Related Concepts

[Understanding Branching](#)
[Overview of Views](#)
[Overview of Branching Options](#)

Related Procedures

[Creating and Configuring Views](#)
[Reviewing or Changing Branching Behavior](#)

Related Reference

[View Type Options and Settings](#)

Reviewing or Changing Branching Behavior

If a folder, file, or change request in a child view has the appropriate settings, you are able to branch it – that is, you can separate it from the corresponding item in the parent view.

At any time, you can determine whether a folder, file, or change request in a branch view is set to **Branch On Change** (that is, its current branching behavior), and if the **Branch On Change** field is enabled, you can change its branching behavior.

Note: Folders and change requests branch when their properties change, while files branch when either their contents or their properties change. Requirements, tasks, and topics never branch

To review or change the branching behavior for a specific file, folder or change request

- 1 Right-click a folder or item and choose **Advanced ► Behavior**.
- 2 Select the **Modify** tab in the **Folder/ Item Behavior** dialog box, and view or modify the **Branch On Change** check box.

Related Concepts

[Overview of Branching Options](#)

[Overview of Views](#)

[Understanding View Types](#)

[Proper Use of Views](#)

[Understanding Branching](#)

Related Procedures

[Managing Views](#)

[Creating and Configuring Views](#)

[Changing a View's Default and Alternate Working Folders](#)

[Copying View Labels](#)

[Deleting Views](#)

[Rolling Back the Current View Configuration](#)

Related Reference

[View Type Options and Settings](#)

Comparing and Merging Views

This section contains procedures related to comparing and merging views. You can use the Cross-Platform Client View Compare/Merge tool, or the [VCMUtility](#) command-line tool to compare and merge views. From Cross-Platform Client View Compare/Merge, you can also open File Compare/Merge tool to compare and merge file contents.

View Compare/Merge is only available in the StarTeam Cross-Platform Client and the command-line [VCMUtility](#).

In This Section

[Changing Access Rights for a Change Package](#)

Describe how to change the access rights for a Change Perspective.

[Changing the Default Merge Type Options and Actions](#)

Describes how to change the default merge type options and actions for a View Compare/Merge merge type.

[Changing the View Compare/Merge Session Working Folder](#)

Describes how to change the temporary default working folder View Compare/Merge uses for a VCM session.

[Changing View Compare/Merge Actions](#)

Describes how to choose an action to resolve merge conflicts in a View Compare/Merge session by changing the action you want applied to the target item.

[Changing View Compare/Merge Perspectives](#)

Describes how to change perspectives in a View Compare/Merge session.

[Cloning a New VCM Merge Type From One Server Configuration to Another](#)

Describes how to clone a newly created VCM merge type from one server configuration to another.

[Committing View Compare/Merge Change Session](#)

Describes how to commit View Compare/Merge change session

[Comparing a Source and Target File in View Compare/Merge](#)

Describes how to compare the contents of two different views using View Compare/Merge.

[Comparing a Target File with the Proposed Merge Results](#)

Describes how to compare the contents of the target file with the proposed merge results in a View Compare/Merge session.

[Comparing and Merging Views](#)

Describes the core procedure for comparing and merging views.

[Comparing and Merging Views](#)

Describes how to compare the contents of two different views using View Compare/Merge.

[Comparing Selected Items in the Source and Target View](#)

Describes how to do a view compare/merge on only selected items from the source and target views.

[Creating a Custom VCM Merge Type](#)

Describes how to create a custom merge type for a View Compare/Merge session.

[Creating a View Compare/Merge Change Package](#)

Describes how to create a View Compare/Merge change package.

[Deleting a Change Package](#)

Describe how to delete a change session.

[Deleting a View Compare/Merge Session](#)

Describes how to delete a View Compare/Merge session.

[Editing a Change Package](#)

Describe how to edit a change packages.

[Editing Change Package Properties](#)

Describe how to edit change package properties.

[Exporting a View Compare/Merge Session](#)

Describes how to export a View Compare/Merge session for reuse later or on a different computer.

[Filtering Items in a View Compare/Merge Session](#)

Describes how to filter items in a View Compare/Merge Session.

[Finding and Breaking Locks on Items with Differences](#)

Describes how to find and break locks on differences.

[Merging Changes Between Any Two Related Views \(Replicate\)](#)

Describes how to merge changes from one view to another (Replicate).

[Merging Changes from a Child View to a Parent View \(Promote\)](#)

Describes how to promote changes from a child view to a parent view (Promote).

[Merging Changes from a Parent View to a Child View \(Rebase\)](#)

Describes how to merge changes from a parent view to a child view (Rebase).

[Naming a Change Package](#)

Describe how to the name of a change package.

[Opening a Change Package in a VCM Session](#)

Describe how to open a change package from the Change Perspective in a VCM session.

[Opening a Change Package in Read-Only Mode for Review](#)

Describe how to open a change package in read-only mode for review in the in the Change Perspective.

[Opening a Saved or Exported View Compare/Merge Session](#)

Describes how to open a previously saved or exported View Compare/Merge session.

[Preparing Your Files for a View Compare/Merge Session](#)

Describes the recommended steps to prepare for a View Compare/Merge session.

[Replaying a Change Package to Another View](#)

Describe how to apply the same changes from a change package to another view..

[Resolving File Differences from a View Compare/Merge Session](#)

Describes how to resolve differences in text file contents in VCM.

[Resolving Merge Conflicts in View Compare/Merge](#)

Describes how to resolve merge conflicts in a View Compare/Merge session.

[Restarting a VCM Session](#)

Describe how to restart a VCM session so you can start over with the same initial session.

[Running View Compare/Merge from the Command-line](#)

Describes how to run the View Compare/Merge command-line utility (`VCMUtility`).

[Saving a VCM Session to a Change Package](#)

Describes how to save a VCM session to a change package.

[Setting Locking Options for a View Compare/Merge Session](#)

Describes how to set locking options for a view merge session.

[Specifying the Initial Perspective for a View Compare/Merge Session](#)

Describes how to specify a starting perspective for a View Compare/Merge session.

[Testing View Compare/Merge Changes](#)

Describes how to use the Test Perspective to test your merge changes.

[Using Filters and Queries to Navigate Change Packages](#)

Describe how to use Filters and Queries in the Change Perspective.

[Using Process Items to Merge Related Files](#)

Describes how to use process items in a View Compare/Merge to select sets of files.

[Viewing and Printing a VCM Session Summary](#)

Describes how to view the View Compare/Merge session properties.

[Viewing and Printing a View Compare/Merge Difference Report](#)

Describes how to view a report on the differences in a View Compare/Merge session.

[Viewing Change Package Details, History and Labels](#)

Describe how to view details, history, and labels about change package sessions in the Change Perspective.

[Viewing Changes in a Change Package](#)

Describe how to view the changes made in a change package session in the Change Perspective.

[Viewing the Change Perspective](#)

Describe how to switch from the standard Content Perspective to the Change Perspective.

Changing Access Rights for a Change Package

Change package access rights are view-level rights. They can be set at the project level as defaults for all views in the project, or for an individual view at the view level.

To change the access rights for a change package

- 1 In the Cross-Platform Client window, open the project or view for which you want to set change package access rights.
- 2 Choose **Project** ► **Access Rights** or **View** ► **Access Rights**.
- 3 Click **Change Package** under the **Categories** list, then continue to specify the access rights for specific groups or people as in changing access rights for other objects.

Note: You can also change the Project or View access rights for a change package from the **Change Perspective**. To do that, click the **Change Perspective** icon, then perform the same steps for changing the access rights as stated above.

Related Procedures

[Assigning Access Rights to Projects](#)

Changing the Default Merge Type Options and Actions

As a server administrator you can modify the availability of default and custom merge types and options when users launch the **VCM** wizard.

To change the default merge type options and actions

- 1 Open the Server Admin tool.
- 2 Click the **Customize VCM** icon in the Administration section.
- 3 In the **Customize VCM** pane, select the merge type in the **All Merge Types** section.
- 4 In the **Available Merge Types** section, select each merge type and click **Add** or **Remove**.
- 5 Click **Save** when you are finished.

Related Concepts

[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Changing the View Compare/Merge Session Working Folder

View Compare/Merge allows you to check out files in a session to a temporary default working folder until you commit your merge changes. This allows you to test your changes to make sure they work before you commit them to the StarTeam repository on the server.

To change the VCM temporary working folder

- 1 In your VCM session, choose **VCM Session** ► **Test Perspective**.
- 2 Choose **VCM Session** ► **Working Folder** to open the **VCM Session Working Folder** dialog box.
- 3 Select **Alternate** and type in, or browse to, the location for your VCM Session working folder.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)
[Changing View Compare/Merge Perspectives](#)
[Comparing and Merging Views](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)
[Conditions for VCM Action Decisions](#)

Changing View Compare/Merge Actions

This procedure describes how to resolve item conflicts in a View Compare/Merge session by changing the action you want applied to the target item. You can change merge actions for items with a status of **Resolved** or **Unresolved**.

To change a View Compare/Merge action

- 1 Start a View Compare/Merge session.
By default, the View Compare/Merge session opens in the **Merge Perspective**.
- 2 Select an item in the upper pane of the **Merge Perspective** that has a status of **Unresolved** or **Resolved**.
- 3 Right-click the selected item and choose an action.

Tip: You can also change the action by using the flyout menu available when you hover the mouse over an icon in the upper pane.

When an unresolved item becomes resolved, View Compare/Merge changes the status to **Resolved**. When all conflicts are resolved, you can commit your changes to the target view.

To view the current merge action for all items in the **Merge Perspective**, display the **Merge Action** column in the upper pane. You can sort the **Merge Action** column, and change the current action for one or more selected items using the context menu or flyout menu.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Testing View Compare/Merge Changes](#)
[Running View Compare/Merge from the Command-line](#)
[Displaying Additional Fields](#)
[Sorting and Grouping Data](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Changing View Compare/Merge Perspectives

When you start a View Compare/Merge session, the VCM Session opens in the **Merge Perspective**.

To change the View Compare/Merge Perspective

- 1 Start a View Compare/Merge session.
- 2 Choose **VCM Session** ▶ **Compare Perspective**, or **VCM Session** ▶ **Test Perspective** to switch to one of those perspectives.
- 3 To switch back to the **Merge Perspective**, choose **VCM Session** ▶ **Merge Perspective**.

Note: You can also use corresponding toolbar buttons to change perspectives.

Related Concepts

[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Cloning a New VCM Merge Type From One Server Configuration to Another

This procedure describes how to clone a newly created VCM merge type from one server configuration to another.

To clone a VCM merge type

- 1 Use the Server Admin tool to create the merge type completely for one server configuration, then click **Save**.
- 2 On the second server configuration, use the Server Admin tool to create the merge type by name only, based on the same VCM default merge type, such as Default.
Click **Save**.
- 3 Use the Server Admin tool to specify which projects use the new merge type and click **Save**.
- 4 From the StarFlow Extensions project on the first server configuration, check out [StarFlow Extensions/Projects/custom.vcm.xml](#),
- 5 Copy the information between `<customVCMTType name=...>` and `</customVCMTType>` for this merge type and save it to the clipboard or another file.
- 6 From the StarFlow Extensions project on the second server configuration, check out [StarFlow Extensions/Projects/custom.vcm.xml](#),
- 7 Replace the information between `<customVCMTType name=...>` and `</customVCMTType>` in this file with the information you saved to the clipboard or file.
- 8 Save the file and check it in.

Note: Cloning information from one [custom.vcm.xml](#) to another is slightly different. It assumes that you have a [custom.vcm.xml](#) in both servers and can paste the sections about merge types from one to the other. However, the XML files also contain information about the project and view structures for each server configuration and this information cannot be cloned. If you simply check the [custom.vcm.xml](#) from server configuration's StarFlow Extensions project into another server configuration's StarFlow Extensions project, you get error messages when you run VCM

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Creating a Custom VCM Merge Type](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Exporting a View Compare/Merge Session](#)
[Deleting a View Compare/Merge Session](#)

Committing View Compare/Merge Change Session

Committing a View Compare/Merge change package session applies all changes to the StarTeam repository. You must resolve all merge conflicts before you can commit the View Compare/Merge change session.

To commit View Compare/Merge change session

- 1 Open the target view in the Cross-Platform Client **Content Perspective** containing the change session you want to commit.
- 2 Double-click the change package you want to commit. It will be listed in the list of change packages in the **Content Perspective** and will have a status of **New**.

Note: If you don't see the session you want, make sure the **Filter** drop-down box has the correct selection so all sessions are displayed.
- 3 Double-click the change package session to open it in the VCM change session.
- 4 Verify that you have resolved all merge conflicts in the **Compare Perspective** and the **Merge Perspective** of a VCM change session, and that you have tested your changes to make sure they give the expected results. When the changes in a change session are complete, it can be committed.
- 5 Choose **VCM Session ► Commit Changes**.

Note: If **Commit** is disabled on the menu or toolbar, check the **Compare** and **Merge Perspectives** to make sure all conflicts are resolved, and items with changes are not locked by other users, then try again.
- 6 Type a merge session comment in the **Commit Changes to View** dialog box.
- 7 Click **OK** In the **Commit Changes to View** dialog box to commit your View Compare/Merge session.
This causes its updates to be applied to the target view in an atomic transaction. In the same transaction, the change package object is updated with details about updated items. After commit, the change session is closed and can no longer be opened for editing. In its place, the committed change package appears as an object owned by the target view, and can be viewed again in the **Change Perspective**.
- 8 Save your change session using **VCM Session ► Save**, **VCMSession ► Save to Local**, or **VCMSession ► Export..**

Note: **Save** saves the session persistently, minimizing the need to save the session as a file. **Save to Local** saves the session as a **.vcms** file, and **Export** saves the session as a **.vcmx** file.

StarTeam creates a uniquely named *Change Package* object in the target view for the change session which contains all the changes made in that VCM change session. It then becomes available in the **Change Perspective**, where it can later be opened in read-only view, reviewed, have the properties modified, or have the session Replayed which allows you to reuse the change package session to apply the same set of changes to another view. Once a change package is committed, no more changes can be made to it.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[View Compare/Merge Actions](#)

[Conditions for VCM Action Decisions](#)

Related Procedures

[Comparing and Merging Views](#)

[Testing View Compare/Merge Changes](#)

[Changing the View Compare/Merge Session Working Folder](#)

[Resolving Merge Conflicts in View Compare/Merge](#)

[Exporting a View Compare/Merge Session](#)

[Saving a VCM Session to a Change Package](#)

[Generating Reports from a File Compare/Merge Session](#)

[Changing View Compare/Merge Actions](#)

[Running View Compare/Merge from the Command-line](#)

Comparing a Source and Target File in View Compare/Merge

In View Compare/Merge, both the **Compare Perspective** and the **Merge Perspective** compare the contents of views. The **Compare Perspective** compares and displays the contents of the source and target views, and provides a list of changes View Compare/Merge will make on commit. The **Merge Perspective** shows items with proposed merge actions, organizing them by folder and filter as in a regular view window. You can also view a comparison of the differences in individual file contents from the **Merge Perspective**. Both perspectives allow you to open File Compare/Merge to resolve and merge the differences.

Note: When you select a file in the **Merge Perspective** and select the tab that compares the target file to the merged file, it is blank if there is no merged file. While the selected file is **Unresolved**, the tab is blank. If the user merges the file or selects a resolving action such as **Overwrite**, then the pane displays content comparing the target file to the content of the resolved file. In the case of the **Ignore** action, the target file is the resolved/merged file. In the case of the **Overwrite** action, the target file is the source file.

To compare a source and target file in the Merge Perspective

- 1 Start a View Compare/Merge session.
By default, the View Compare/Merge window opens in the **Merge Perspective**. Stay in this perspective.
- 2 Select a file with differences, having a merge status of either **Unresolved** (question mark), or **Resolved** (check mark).
- 3 Click the **File Comparison: Source, Target** tab at the bottom.

You can only compare the file differences in View Compare/Merge. To merge the differences, you can open File Compare/Merge.

Note: Color highlighting of differences in the **Compare** pane are explained and controlled in the **File Compare/Merge Options** dialog box. To change the default highlighting colors for file and folder comparison differences for both View Compare/Merge and File Compare/Merge, modify the File Comparison and Folder Comparison Color Preferences accessed by choosing **Tools ▸ Options** in the File Compare/Merge stand-alone window.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Resolving File Differences from a View Compare/Merge Session](#)
[Using Process Items to Merge Related Files](#)
[Saving a VCM Session to a Change Package](#)
[Generating Reports from a File Compare/Merge Session](#)
[Changing View Compare/Merge Actions](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Comparing a Target File with the Proposed Merge Results

When you are in a View Compare/Merge session, you can compare the text contents of the target file with the proposed merge results using the **Merge Perspective**.

To compare the contents of a target file with the proposed merged results

- 1 Start a View Compare/Merge session.
By default, the View Compare/Merge window opens in the **Merge Perspective**. Stay in this perspective.
- 2 Select a file with differences (having a merge status of either **Resolved** or **Unresolved**).
- 3 Click the **File Comparison: Target, Merged** tab at the bottom of the lower pane.

Note: This pane only allows you to compare the file differences. To merge the differences, you can open File Compare/Merge.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Resolving File Differences from a View Compare/Merge Session](#)
[Using Process Items to Merge Related Files](#)
[Changing View Compare/Merge Actions](#)

Comparing and Merging Views

This core procedure presents the basic high-level tasks involved in performing a view compare/merge in StarTeam.

To compare and merge views

- 1 Prepare your files for a View Compare/Merge session.
[Preparing Your Files for a View Compare/Merge Session](#)
- 2 Start a View Compare/Merge session using the **View Compare/Merge Wizard**.
[Comparing and Merging Views](#)
- 3 Resolve merge conflicts between the views.
[Resolving Merge Conflicts in View Compare/Merge](#)
- 4 Change any actions proposed by View Compare/Merge that you consider inappropriate for this session.
[Changing View Compare/Merge Actions](#)
- 5 Test the merge changes to make sure they are all correct.
[Testing View Compare/Merge Changes](#)
- 6 Generate a View Compare/Merge session report of all the merge changes.
[Viewing and Printing a View Compare/Merge Difference Report](#)
- 7 Commit the merge changes.

Note: If you need to stop work on the VCM change session before resolving and committing the changes, or you want to finish it later after having someone else review the proposed changes, you can use one of the **VCM Session** menu items **Save**, **Save to Local**, or **Export** before committing the session. StarTeam creates a *Change Package* object with a unique name in the target view which contains all the changes proposed for that VCM change session. It is then available in the **Change Perspective** where it can later be restarted, reviewed, committed, and once committed, have the session Replayed which reuses the change package session to apply the same set of changes to another view.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Preparing Your Files for a View Compare/Merge Session

Before starting a View Compare/Merge session, take the following steps to prepare your files.

Note: To compare or merge specific items in a view, or all items in specific folders in the view, start the View Compare/Merge session from the **Advanced** context menu on the selected items, rather than from the **View** menu. When you start a View Compare/Merge session from the **View** menu, all items of requested types and folders in the view are included.

To prepare your files for a view compare/merge session

- 1 Check in any modified files that will be part of the View Compare/Merge session, and optionally create a view or revision label to use for the source.
- 2 Select the views you want to compare and set their properties to permit items to branch on change.

Note: The target view is the only view that receives the results of the merge.

- 3 Decide whether you want to merge all changes in a view, or merge changes to selected items in a view. Select individual items in the upper pane if you are including specific items in the View Compare/Merge session. You can select multiple items on multiple component tabs, including the **Folder** tab.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Merging Changes from a Child View to a Parent View \(Promote\)](#)
[Merging Changes from a Parent View to a Child View \(Rebase\)](#)
[Merging Changes Between Any Two Related Views \(Replicate\)](#)
[Using Process Items to Merge Related Files](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Comparing and Merging Views

Use the **Compare Perspective** and the **Merge Perspective** in a View Compare/Merge session (VCM session) to compare the contents of a source and target view, and to review or change the proposed merge actions for resolving item differences.

Note: The steps and options included in the **View Compare/Merge Wizard** differ depending on which compare or merge type you select on the first page of the **View Compare/Merge Wizard**.

To compare views

- 1 Choose **Project** ► **Open** and in the **Open Project Wizard**, select the project and one of the views you want to compare. Click **Finish**.
- 2 In the opened view, select the root folder in the folder tree to compare items in the entire view, or select one of the sub folders to just compare items from that folder.
- 3 Right-click the selected folder and choose **Advanced** ► **View Compare/Merge** to open the **View Compare/Merge Wizard**.
- 4 Select the type of compare or merge you want to perform on the first page of the **View Compare/Merge Wizard**: **Compare only**, **Promote**, **Rebase**, or **Replicate**.

The **View Compare/Merge Wizard** uses the configuration to which the view is currently set to determine which choices to present on the first page of the wizard.

Note: If you chose to perform a **Compare Only** merge of two reference views, on the **Select Source/Target Configuration** step of the **View Compare/Merge Wizard**, select **Current Configuration**. Do not select any of the other types of configurations for the **Compare Only** merge.

- 5 Click **Next** and continue through the wizard selecting the options you want until you get to the **Set Options** page of the wizard.

Optionally, uncheck the **Start With Merge Perspective Current** checkbox option on the **Set Options** page of the wizard if you want to start the VCM session in the **Compare Perspective**.

This option is set to **Start With Merge Perspective Current** by default which causes the VCM session to open in the **Merge Perspective**.

Select any other options you want and click **Next**.

Note: You can change perspectives as needed in the VCM session to accomplish different tasks.

- 6 The **Exclude Properties** page appears next and gives you the opportunity to exclude any properties for each branchable item type for which you do not want to merge changes when the session is committed.

Select an item type in the **Item Type** list, then select a property for that item in the **Available properties** list that you want to exclude from the commit. Click the **Exclude** button to move it to the **Exclude properties** list.

Note: If you change your mind, you can always select the item again, select the excluded property, and click **Include** to return it to the **Available Properties** list.

- 7 Click **Finish**.

This opens the View Compare/Merge session in either the **Merge Perspective** or **Compare Perspective**.

- 8 Use the **Merge Perspective** or **Compare Perspective** to change merge actions and resolve item differences.

Tip: You can switch to the **Test Perspective** to see what the target view would look like if you applied your changes. Switch back to the **Merge Perspective** or **Compare Perspective** to make additional changes as needed.

Note: If you use need to stop work on the VCM change session before resolving and committing the changes, or you want to finish it later after having someone else review the proposed changes, you can use one of the **VCM Session** menu items **Save**, **Save to Local**, or **Export** before committing the session. StarTeam creates a uniquely named *Change Package* object in the target view which contains all the changes proposed for that VCM change session. It is then available in the **Change Perspective** where it can later be restarted, reviewed, committed, and once committed, have the change session Replayed which reuses the change package session to apply the same set of changes to another view.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)
[View Compare/Merge Session Perspectives](#)
[Conditions for VCM Action Decisions](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Changing View Compare/Merge Actions](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Testing View Compare/Merge Changes](#)
[Resolving File Differences from a View Compare/Merge Session](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Resolving Merge Conflicts in View Compare/Merge

You can resolve merge conflicts in the **Merge Perspective** or the **Compare Perspective** of a View Compare/Merge session. You can also open File Compare/Merge directly from the View Compare/Merge session to resolve differences in text file contents.

Note: When merging views, no **Merge** or **Mark As Resolved** actions are possible for nonbranching items, which are requirements, tasks, and topics.

To resolve merge conflicts in a View Compare/Merge session

- 1 Choose **VCM Session** ► **Merge Perspective** or **VCM Session** ► **Compare Perspective** to switch to one of these two perspectives.
- 2 Double-click an item with an **Unresolved**, **Resolved**, or **Merge** status.

Note: The item status displays in the **Merge Action** column of the **Merge Perspective**, or in the **Action** column of the **Compare Perspective**.

The **Resolve/Merge Properties** dialog box opens displaying the item properties.

- 3 Select a property with a conflict to resolve.

Note: Properties with conflicts display in **boldface** type.

- 4 Click the **Use Source**, **Use Target**, or **Merge** button.

Tip: To quickly specify an action, double-click the **Source** or **Target** column for the property to use that value in the **Merged** column.

- 5 Optionally, double-click the **Merged** column on a text file to resolve the text differences manually.

When an unresolved item is resolved, View Compare/Merge updates the status to **Resolved** and changes the status icon to a check mark.

Tip: You can also resolve conflicts directly in the **Property** pane of the **Merge Perspective**. Right-click a property and choose an action from the context menu.

Note: If you use need to stop work on the VCM change session before resolving and committing the changes, or you want to finish it later after having someone else review the proposed changes, you can use one of the **VCM Session** menu items **Save**, **Save to Local**, or **Export** before committing the session. StarTeam creates a *Change Package* object with a unique name in the target view which contains all the changes proposed for that VCM session. It is then available in the **Change Perspective** where it can later be restarted, reviewed, committed, and once committed, have the change package replayed, which reuses the change package to apply the same set of changes to another view.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[View Compare/Merge Actions](#)

[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)

[Resolving File Differences from a View Compare/Merge Session](#)

[Testing View Compare/Merge Changes](#)

[Saving a VCM Session to a Change Package](#)

[Changing View Compare/Merge Actions](#)

[Running View Compare/Merge from the Command-line](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Changing View Compare/Merge Actions

This procedure describes how to resolve item conflicts in a View Compare/Merge session by changing the action you want applied to the target item. You can change merge actions for items with a status of **Resolved** or **Unresolved**.

To change a View Compare/Merge action

- 1 Start a View Compare/Merge session.
By default, the View Compare/Merge session opens in the **Merge Perspective**.
- 2 Select an item in the upper pane of the **Merge Perspective** that has a status of **Unresolved** or **Resolved**.
- 3 Right-click the selected item and choose an action.

Tip: You can also change the action by using the flyout menu available when you hover the mouse over an icon in the upper pane.

When an unresolved item becomes resolved, View Compare/Merge changes the status to **Resolved**. When all conflicts are resolved, you can commit your changes to the target view.

To view the current merge action for all items in the **Merge Perspective**, display the **Merge Action** column in the upper pane. You can sort the **Merge Action** column, and change the current action for one or more selected items using the context menu or flyout menu.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Testing View Compare/Merge Changes](#)
[Running View Compare/Merge from the Command-line](#)
[Displaying Additional Fields](#)
[Sorting and Grouping Data](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Testing View Compare/Merge Changes

To test proposed changes in a View Compare/Merge session

- 1 Verify that you have resolved all merge conflicts in the **Compare Perspective** or **Merge Perspective** of the View Compare/Merge session.

Note: You cannot check out individual files with merge conflicts in the **Test Perspective** of a View Compare/Merge session. Resolve all conflicts before using the **Test Perspective**.

- 2 Choose **VCM Session** ► **Test Perspective** and check out the files in your project, including those with merged changes.

View Compare/Merge saves all files locally to a temporary default working folder until you commit the View Compare/Merge session.

Tip: You can specify an alternate View Compare/Merge working folder. Choose **VCM Session** ► **Working Folder** in the **Test Perspective**.

- 3 Use your build tools to test that the project will build, run, and has the needed changes.

Note: You cannot build or run from within StarTeam.

- 4 Optionally, switch to the **Merge Perspective** or the **Compare Perspective** and make changes as needed.
- 5 Recheck out your project if you have made additional changes, and rerun your tests.
- 6 Choose **VCM Session** ► **Commit Changes** when you are satisfied with all the proposed changes.

View Compare/Merge applies all the changes to the StarTeam server repository and closes the session.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)
[Changing the View Compare/Merge Session Working Folder](#)
[Comparing and Merging Views](#)
[Changing View Compare/Merge Actions](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)
[Conditions for VCM Action Decisions](#)

Viewing and Printing a View Compare/Merge Difference Report

At any time during a View Compare/Merge session, you can view a report of the differences in the session to see what you have done and what you have left to do. There are two types of difference reports available:

- ◆ **Difference Report:** Displays information about the VCM session parameters and options, a summary of the number of differences by type and action, and the view differences between the source and target by file and folder.
- ◆ **Detailed Difference Report:** Displays the same information as the **Difference Report**, but also includes information about items whose parent folder has moved in the source or target view. If no folders were moved, this report will look the same as the **Difference Report**.

To view and print a Difference Report for a View Compare/Merge session

- 1 Start a View Compare/Merge session, or open a saved change package from the **Change Perspective** using **Change Package ▶ Open (in VCM)**, or **Change Package ▶ Open as Read Only (in VCM)**.
- 2 Choose **VCM Session ▶ Difference Report** or **VCM Session ▶ Detailed Difference Report**.
The **VCM Session Differences** dialog box displays showing a two-column list of the session differences.
- 3 Optionally, click the **Browse** button to choose a different path and name for the output filename.
The report is saved as an `.html` file.
- 4 Press the **View In Browser** button to display the difference report in your default browser.
- 5 Use the default browser menus to print the report.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing and Merging Views](#)
[Changing the View Compare/Merge Session Working Folder](#)
[Testing View Compare/Merge Changes](#)
[Saving a VCM Session to a Change Package](#)
[Exporting a View Compare/Merge Session](#)
[Viewing and Printing a VCM Session Summary](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Comparing and Merging Views

Use the **Compare Perspective** and the **Merge Perspective** in a View Compare/Merge session (VCM session) to compare the contents of a source and target view, and to review or change the proposed merge actions for resolving item differences.

Note: The steps and options included in the **View Compare/Merge Wizard** differ depending on which compare or merge type you select on the first page of the **View Compare/Merge Wizard**.

To compare views

- 1 Choose **Project** ► **Open** and in the **Open Project Wizard**, select the project and one of the views you want to compare. Click **Finish**.
- 2 In the opened view, select the root folder in the folder tree to compare items in the entire view, or select one of the sub folders to just compare items from that folder.
- 3 Right-click the selected folder and choose **Advanced** ► **View Compare/Merge** to open the **View Compare/Merge Wizard**.
- 4 Select the type of compare or merge you want to perform on the first page of the **View Compare/Merge Wizard**: **Compare only**, **Promote**, **Rebase**, or **Replicate**.

The **View Compare/Merge Wizard** uses the configuration to which the view is currently set to determine which choices to present on the first page of the wizard.

Note: If you chose to perform a **Compare Only** merge of two reference views, on the **Select Source/Target Configuration** step of the **View Compare/Merge Wizard**, select **Current Configuration**. Do not select any of the other types of configurations for the **Compare Only** merge.

- 5 Click **Next** and continue through the wizard selecting the options you want until you get to the **Set Options** page of the wizard.

Optionally, uncheck the **Start With Merge Perspective Current** checkbox option on the **Set Options** page of the wizard if you want to start the VCM session in the **Compare Perspective**.

This option is set to **Start With Merge Perspective Current** by default which causes the VCM session to open in the **Merge Perspective**.

Select any other options you want and click **Next**.

Note: You can change perspectives as needed in the VCM session to accomplish different tasks.

- 6 The **Exclude Properties** page appears next and gives you the opportunity to exclude any properties for each branchable item type for which you do not want to merge changes when the session is committed.

Select an item type in the **Item Type** list, then select a property for that item in the **Available properties** list that you want to exclude from the commit. Click the **Exclude** button to move it to the **Exclude properties** list.

Note: If you change your mind, you can always select the item again, select the excluded property, and click **Include** to return it to the **Available Properties** list.

- 7 Click **Finish**.

This opens the View Compare/Merge session in either the **Merge Perspective** or **Compare Perspective**.

- 8 Use the **Merge Perspective** or **Compare Perspective** to change merge actions and resolve item differences.

Tip: You can switch to the **Test Perspective** to see what the target view would look like if you applied your changes. Switch back to the **Merge Perspective** or **Compare Perspective** to make additional changes as needed.

Note: If you use need to stop work on the VCM change session before resolving and committing the changes, or you want to finish it later after having someone else review the proposed changes, you can use one of the **VCM Session** menu items **Save**, **Save to Local**, or **Export** before committing the session. StarTeam creates a uniquely named *Change Package* object in the target view which contains all the changes proposed for that VCM change session. It is then available in the **Change Perspective** where it can later be restarted, reviewed, committed, and once committed, have the change session Replayed which reuses the change package session to apply the same set of changes to another view.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)
[View Compare/Merge Session Perspectives](#)
[Conditions for VCM Action Decisions](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Changing View Compare/Merge Actions](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Testing View Compare/Merge Changes](#)
[Resolving File Differences from a View Compare/Merge Session](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Comparing Selected Items in the Source and Target View

To compare selected items in the views

- 1 Open the source or target view you want to use in your compare/merge session.
- 2 Choose **View** ► **Configuration** to open the **View Configuration** dialog box, and choose a configuration unless you plan to use the current one.

Note: The **VCM Wizard** uses the configuration to which the view is currently set if you start the wizard from the source view.

- 3 Select the individual items to compare on the component tabs in the upper pane, or select a folder on the **Folder Tree** in the left pane.

Note: The items you select on one tab will stay selected when you switch component tabs.

- 4 Right-click the selected items and choose **Advanced** ► **View Compare/Merge**.
This opens the **View Compare/Merge Wizard**.
- 5 On the first page of the **View Compare/Merge Wizard**, select the type of compare or merge you want to perform.
- 6 Click **Next** and continue through the wizard selecting the options you want.

Note: The remaining steps and options in the wizard differ depending on which compare or merge type you selected.

- 7 On the **Set Options** page of the wizard, uncheck **Start With Merge Perspective Current**.
Leaving this option unchecked causes the View Compare/Merge window to open in the **Compare Perspective**.

The upper pane displays the contents of the source and target views. It highlights items with no matches in green, matched items with differences in yellow, and items moved to another folder in blue. You open, close, and navigate folders. Each comparison pane contains scroll bars and navigation buttons. Selecting an item in the top panes also selects it in the lower pane. Use filters to control which items display in the upper pane.

The lower pane displays information about the items in the upper pane. You can modify the merge action for each item and resolve differences.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)
[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)
[Opening a Saved or Exported View Compare/Merge Session](#)
[Merging Changes from a Child View to a Parent View \(Promote\)](#)
[Merging Changes from a Parent View to a Child View \(Rebase\)](#)
[Merging Changes Between Any Two Related Views \(Replicate\)](#)
[Using Process Items to Merge Related Files](#)
[Changing the View Compare/Merge Session Working Folder](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Creating a Custom VCM Merge Type

As a server administrator, you can create new, custom merge types for View Compare/Merge based on the standard merge types of Promote, Rebase, Replicate, or Compare.

Note: Before you can create a custom VCM merge type, the server must have a StarFlow Extensions project. First create a StarFlow Extensions project, then create the `PROJECTS` folder under the root folder in the view. Otherwise the save operation fails in the **Customize VCM** tool.

To create a custom merge type for VCM

- 1 Open the Server Administration tool and click the **Customize VCM** icon in the **Administration** section.
- 2 In the **Customize VCM** pane, select one of the standard merge types in the **All Merge Types** section.
- 3 Click the **New Custom Merge Type** icon (the red asterisk in the left margin of the **All Merge Types** section.)

Tip: You can create a new merge type by copying an existing custom merge type. Select the custom merge type in the list and click **Copy Custom Merge Type**.

Clicking **New Custom Merge Type** or **Copy Custom Merge Type** displays the customization options in the lower area of the **Customize VCM** pane.

- 4 Give the new merge type a **Name** and **Parent Type**.

Note: All custom merge types must have a **Name** and a **Parent Type** because they are all derived from either Promote, Rebase, Replicate, or Compare.

- 5 In the **Available Merge Types** section, select the context level in the tree to which you want this merge type to be added, and then click **Add**.

Note: The **Available Merge Types** tree is hierarchical. When you add a merge type to a node, it becomes available at all the child node levels under it. To make the new merge type available at all levels, add it to the **Server** node at the top of the tree.

Tip: You can change the order in which these merge types will display in the **View Compare/Merge** wizard. Select a merge type in the tree and click the **Up** or **Down** arrow on the right to move it.

- 6 Enter the description you want to display in the **View Compare/Merge** wizard.
- 7 Choose the default **Merge Action** you want for each **Difference Type**.

The default merge actions are used in the compare phase, are not visible to the user in the **View Compare/Merge** wizard, and the user cannot change them.

Note: A merge action that has been changed from the default parent action is displayed with red text.

- 8 In the **Include Item Types** section, optionally check which item types to pre-select for this merge type. The user can view and change these types on the **Include Selected Items** page of the **View Compare/Merge** wizard.

- 9 In the **Options** section, optionally check which options you want to pre-select for this merge type. The options selected are performed by default when the VCM session begins the compare phase. The user can view and change these options on the **Set Options** page of the **View Compare/Merge** wizard.

10 Click **Save** when you are finished.

The new merge type now appears in the **Available Merge Types** list of the **Customize VCM** tool.

Related Procedures

[Comparing and Merging Views](#)

[Cloning a New VCM Merge Type From One Server Configuration to Another](#)

[Comparing and Merging Views](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Creating a View Compare/Merge Change Package

To create a change package, you simply start a View Compare/Merge (VCM) change session.

To create a change package

- 1 Open one of the two project views you want to compare in the StarTeam client.
- 2 Follow the steps in the procedure “Comparing and Merging Views”.
[Comparing and Merging Views](#)
- 3 Commit your change package session
- 4 Save your change package session.
[Saving a VCM Session to a Change Package](#)

Note: If you use one of the VCM menu **VCM Session** menu items **VCM Session**, or **Export** after committing the session, StarTeam creates a uniquely named *Change Package* object in the target view for the change session which contains all the changes made in that VCM change session. It then becomes available in the **Change Perspective**, where it can later be opened in read-only view, reviewed, have the properties modified, or have the change session Replayed which reuses the change package to apply the same set of changes to another view. Once a change package is committed, no more changes can be made to it.

You can also use **VCM Session Save**

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)

Related Reference

[View Compare/Merge Wizard](#)

Comparing and Merging Views

This core procedure presents the basic high-level tasks involved in performing a view compare/merge in StarTeam.

To compare and merge views

- 1 Prepare your files for a View Compare/Merge session.
[Preparing Your Files for a View Compare/Merge Session](#)
- 2 Start a View Compare/Merge session using the **View Compare/Merge Wizard**.
[Comparing and Merging Views](#)
- 3 Resolve merge conflicts between the views.
[Resolving Merge Conflicts in View Compare/Merge](#)
- 4 Change any actions proposed by View Compare/Merge that you consider inappropriate for this session.
[Changing View Compare/Merge Actions](#)
- 5 Test the merge changes to make sure they are all correct.
[Testing View Compare/Merge Changes](#)
- 6 Generate a View Compare/Merge session report of all the merge changes.
[Viewing and Printing a View Compare/Merge Difference Report](#)
- 7 Commit the merge changes.

Note: If you need to stop work on the VCM change session before resolving and committing the changes, or you want to finish it later after having someone else review the proposed changes, you can use one of the **VCM Session** menu items **Save**, **Save to Local**, or **Export** before committing the session. StarTeam creates a *Change Package* object with a unique name in the target view which contains all the changes proposed for that VCM change session. It is then available in the **Change Perspective** where it can later be restarted, reviewed, committed, and once committed, have the session Replayed which reuses the change package session to apply the same set of changes to another view.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Saving a VCM Session to a Change Package

There are several ways to save a VCM session: **VCM Session ▸ Save**, and **VCM Session ▸ Save to Local**. However, only **VCM Session ▸ Save** saves the session as a change package.

Note: To open a VCM session on a different computer, export the file instead of saving it by using **VCM Session ▸ Export** menu. **Export** creates a VCM exchange file (.vcmx) which allows the entire VCM session to be transported to another computer, allowing the session to be imported from the **Project ▸ Open Shortcut** menu on that computer.

To save a View Compare/Merge Session

- 1 Start a View Compare/Merge Session and resolve the conflicts in it.
- 2 Choose **VCM Session ▸ Save**.
The **Change Package Properties** dialog box opens.
- 3 Make any desired changes to the change package properties and click **Save**.
This saves the VCM Session as a change package object into the target view of the StarTeam Server.

Once you have saved your VCM session, you can continue editing your session until you are ready to save it again or commit it.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Exporting a View Compare/Merge Session](#)

[Opening a Change Package in a VCM Session](#)

[Opening a Saved or Exported View Compare/Merge Session](#)

Deleting a Change Package

Prior to being committed, a change package object can be deleted. As with all versioned objects, the delete is “soft”, meaning the change package will no longer appear in the tip revision of the target view. However, if the target view is rolled back in time, a previously deleted change package will “reappear” and can be opened in read-only mode.

There is no way to “undelete” a change package. After a change package is committed, it cannot be directly deleted. Committed change packages are deleted only when the target view is deleted. Database information no longer needed by a deleted change package is removed during the database purge process.

Deleting a change package also causes it to be closed in the Cross-Platform Client. If any items were locked in the target view on behalf of the change package, the user is prompted as to whether to retain those locks or release them.

Note: Each change package from a VCM session is saved in the target view.

To delete a change package

- 1 In the Cross-Platform Client window, open the target view containing the change package you want to delete.
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 In the list of change packages, click the change package you want to delete.
- 4 Choose **Change Package** ► **Delete**, or right-click the selected package and choose **Delete**.

This action removes the `.vcms` and `.vcmx` session files that were previously saved locally or on the server.

Related Concepts

[Change Packages Overview](#)
[Change Perspective UI](#)

Related Procedures

[Editing a Change Package](#)

Deleting a View Compare/Merge Session

You can delete an active View Compare/Merge (VCM) session which removes all session information from your computer.

To delete a VCM session

- 1 Choose **VCM Session** ► **Delete**.
- 2 Click **Yes** or **No**.

Yes deletes the active session in the client, the session shortcut file (.vcms) and the session repository folder.

No closes the active session in the client, but does not delete the shortcut file or session repository.

Note: When you normally close a VCM session, the session is lost unless it was previously saved, or you modify it and save it when closing.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

[Testing View Compare/Merge Changes](#)

[Viewing and Printing a View Compare/Merge Difference Report](#)

[Saving a VCM Session to a Change Package](#)

[Exporting a View Compare/Merge Session](#)

[Opening a Saved or Exported View Compare/Merge Session](#)

Editing a Change Package

You can only edit a change package with a status of **New** in the **Change Perspective**, which means it has been saved but not yet committed. This procedure presumes you have already saved one or more uncommitted change packages which will be visible in **Change Perspective** of the target view in which it resides.

When a change package is opened for editing, the only item-level changes allowed are changing VCM action types such as marking a difference as **ignore**, and resolving merge conflicts, for example, manually merging properties or file contents.

The only non-item edits allowed are: the change package's **Name** and **Description** properties, **Pre-and Post-Comment Revision** and **View Label** options, the change package's **Revision Comment** which is used when the change package is saved or committed, the change package's **Working Folder**, and the change package's **Default Comment** which is used as the revision comment for any new item revisions created during commit.

To edit a change package

- 1 In the Cross-Platform Client window, open the target view of the change package you want to use.
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 In the list of change packages, select the uncommitted change package you want to edit and choose **Change Package** ► **Open (in VCM)** to reopen the change package as an active VCM session.
Alternatively, you can double-click the change package and it will automatically open it in the VCM session.
- 4 In the VCM session, make the changes you want.
- 5 Save or commit the changed session when you are finished.

Related Concepts

[Change Packages Overview](#)
[Change Perspective UI](#)

Related Procedures

[Editing Change Package Properties](#)
[Replaying a Change Package to Another View](#)

Editing Change Package Properties

A selected change package's properties can be viewed in the **Change Perspective**, but when the change package is re-opened in the VCM session, its properties can be edited with the **VCM Session Properties** dialog box. This dialog box serves several purposes. It allows you not only to view or change the properties of the change package, but it also allows you to commit the changes if the session was new, add or change the change session labels, and add a comment for the committed revision.

This procedure presumes you have already saved one or more change packages that will be visible in **Change Perspective** of the target view in which it resides.

Note: Each change package is saved in the target view.

To edit change package properties

- 1 In the Cross-Platform Client window, open the target view of the change package you want to use.
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.

Note: If the change package is committed it cannot be edited further.

- 3 Select the change package on the left and double-click it to open it in the VCM session.
You can always view session properties by choosing **Change Package** ► **Properties**, but the properties are read-only in this case.
- 4 Choose **VCM Session** ► **Properties** to open the **Change Session Properties** dialog box.
- 5 Modify the properties desired, then click **OK**.

Note: If the session was uncommitted, you can change any of the properties for the session. However, if the session was committed before you reopened it in the VCM session, the only properties you will be able to change are the **Default** or **Alternate** working folder.

Related Concepts

[Change Packages Overview](#)
[Change Perspective UI](#)

Related Procedures

[Editing a Change Package](#)
[Replaying a Change Package to Another View](#)

Exporting a View Compare/Merge Session

Before you leave a View Compare/Merge (VCM) session, you can export it to a file so you can open the session again later, or transfer the session to a different computer. View Compare/Merge saves all the session information in a <VCM exchange file> file with a `.vcmx` extension to which you can send a StarTeam shortcut URL to others to open, or which can be sent to others for review, or opened from the **Change Perspective**

Note: If you only expect to open the session later on the same machine, to reduce the file size, you can use **VCM Session ► Save**, or **VCM Session ► Save to Local**. The `.vcms` file created with a **Save to Local** only saves part of the session information. Data from the local repository directory is not included in this file. This format is used to automatically save data when closing StarTeam with a VCM session open and only allows you to open it on the machine on which it was saved.

To export a VCM session

- 1 Choose **VCM Session ► Export** on the View Compare/Merge menu.
This opens the **Export VCM Session As** file browser.
- 2 Make sure that the **Files of Type** field has `VCM Session Files (*.vcmx)` selected.
- 3 Browse to the target location for the `.vcmx` file.
- 4 Optionally, give the file a different name. You can also specify the name and location for a shortcut file.

Note: The default name consists of the type of compare/merge it is, followed by the date and time of the session.

- 5 Click **Save** to export the session to the `.vcmx` file.

Note: An exported VCM change session will be displayed in the **Change Perspective** of the Cross-Platform Client. If it was uncommitted before it was saved, you can restart it in the **Change Perspective**, review and make any changes, then commit the change session.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)
[Opening a Saved or Exported View Compare/Merge Session](#)
[Saving a VCM Session to a Change Package](#)
[Comparing and Merging Views](#)
[Changing the View Compare/Merge Session Working Folder](#)
[Testing View Compare/Merge Changes](#)
[Generating Reports from a File Compare/Merge Session](#)
[Deleting a View Compare/Merge Session](#)

Filtering Items in a View Compare/Merge Session

You can use filters in a View Compare/Merge session to limit the data shown on the upper pane, just as you do in a normal view in the StarTeam client. StarTeam provides a specific set of View Compare/Merge fields and predefined filters to use in View Compare/Merge sessions. You can also add new filters based on View Compare/Merge fields.

To use a predefined filter

- 1 Click a component tab in the upper pane of the **Merge Perspective** or **Test Perspective**, or switch to the **Compare Perspective**.
- 2 Select a filter from the filter drop-down list on the toolbar.

Each perspective contains a different list of filters. Also, there are two filter lists in the **Compare Perspective**: one for item types, and one for status and merge actions.

Note: Only filters containing View Compare/Merge fields can be used in the **Merge Perspective**. View Compare/Merge fields are: **Default Action**, **Difference Type**, **Is Action Overridden?**, **Merge Action**, and **Merge Status**.

To save the current arrangement as a filter

- 1 Click a component tab in the upper pane of the **Merge Perspective** or **Test Perspective**.
- 2 Optionally, right-click a column header, choose **Show Fields**, and add any missing fields to the upper pane you want to include in your filter.

Note: Be sure to include at least one View Compare/Merge field if you want the filter to be used in the **Merge Perspective**.

- 3 Sort and group the data shown on the upper pane, as desired.
- 4 Right-click a column header in the upper pane and choose **Save Current Settings**.
The **Save Current Settings** dialog box opens.
- 5 Type a name for this filter in the **Filter Name** text box.
- 6 Check **Public** to add this filter to the project so anyone with the appropriate access rights can use it, or uncheck **Public** to make the filter private (available only to your user ID).
- 7 Click **OK**.

Note: Creating a new filter in a View Compare/Merge session is the same as in the main StarTeam client window, except that you need to ensure at least one View Compare/Merge field is included for a **Merge Perspective** filter.

Related Concepts

[Filters](#)

Related Procedures

[Creating Filters](#)

[Applying Predefined Filters](#)

[Copying Filters](#)

[Component-level Filter Access Rights](#)

[Individual Filter Access Rights](#)

[Displaying Additional Fields](#)

[Sorting and Grouping Data](#)

Finding and Breaking Locks on Items with Differences

If an item with differences is locked by another user in the target view for the merge, you will be unable to commit changes for that item in the View Compare/Merge (VCM) session unless you break the lock. When you start a VCM session, the **View Compare/Merge Wizard** can find the locks and break them automatically if you check that option in the wizard. However, you can also break the locks manually in the VCM session.

Warning: Items are typically locked with exclusive locks because they are being edited. If you break another user's lock, you may cause that user to lose work. You should break another user's locks only after contacting them and in urgent cases where they cannot be contacted.

To break locks automatically when starting a VCM session

- 1 Start your VCM session using the **View Compare/Merge Wizard**.
- 2 On the **Set Options** page, check **Break Locks Automatically**.
- 3 Check any other options you want, then click **Finish** to complete the wizard.

To quickly find locked items with differences

- 1 Start your VCM session.
- 2 In any perspective, choose **Difference** ► **Find Next Locked**, or **Difference** ► **Find Previous Locked**.

To break existing locks in a VCM session

- 1 Find the lock as described above.
- 2 Right-click the item in the upper pane and choose **Lock/Unlock**.
This opens the **Set My Lock Status** dialog box.
- 3 Select the **Unlocked** lock status option to remove the exclusive or non-exclusive lock.
- 4 Check **Break Existing Lock** to break another team member's lock on the item.

If e-mail is enabled, StarTeam will send an e-mail message to the team member whose lock has been broken to inform him or her of this fact.

Note: You must be granted the appropriate privileges to be able to break another user's locks.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[View Compare/Merge Actions](#)

[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing Selected Items in the Source and Target View](#)

[Resolving File Differences from a View Compare/Merge Session](#)

[Changing View Compare/Merge Actions](#)

Related Reference

[Conditions for VCM Action Decisions](#)

[View Compare/Merge Wizard](#)

Merging Changes Between Any Two Related Views (Replicate)

A Replicate View Compare/Merge session compares or merges changes between any two views that are in the same project. You can start a Replicate View Compare/Merge session from either the source view or the target view.

To perform a Replicate operation using the current view as the source

- 1 Open the view you want to use as the source in your View Compare/Merge session.
- 2 Choose **View** ► **Select Configuration** and specify the source view configuration you want to use by selecting the appropriate configuration type: **Current**, **Labeled**, **Promotion State**, or **Configuration As Of**.

Note: The **View Compare/Merge Wizard** uses the configuration to which the view is currently set if you start the wizard from the source view.

- 3 Choose **View** ► **Compare/Merge**, or right-click selected items in the upper pane and choose **Advanced** ► **View Compare/Merge**.

This opens the **View Compare/Merge Wizard**.

- 4 Select **Replicate** as the compare/merge type.
- 5 Select **Source of Merge** under **Use Current View (name) For**, and click **Next**
A **View Tree** displays all the views in the project.
- 6 Select the target view for the comparison and click **Next**.
- 7 Check one or more types of items you want to include in the compare/merge session, and click **Next**.
- 8 Check one or more compare/merge options to use at the beginning of the compare/merge.
- 9 Click **Finish**.

View Compare/Merge begins comparing the views immediately and opens a new StarTeam window in the **Merge Perspective** where you can resolve any remaining merge conflicts. Items with unresolved differences are easy to find. In the **Merge Status** column for each of these items, View Compare/Merge displays a question mark, followed by the word **Unresolved**. The component tabs and folders also display a question mark if they contain items with unresolved differences.

You must resolve each of these merge conflicts to complete the merge and commit the changes.

To perform a Replicate operation using the current view as the target

- 1 Open the view and configuration you want to use as the target in your View Compare/Merge session.
- 2 Choose **View** ► **Compare/Merge**, or right-click selected items in the upper pane and choose **Advanced** ► **View Compare/Merge**.
- 3 Select **Replicate** as the compare/merge type.
- 4 Select **Target Of Merge** under **Use Current View (name) For**, and click **Next**
- 5 Select the view in the **View Tree** to use as the source of the merge.
- 6 Choose the source view configuration to use: **Current**, **Labeled**, **Promotion State**, or **Configuration As Of** (a specific timestamp).
Click **Next**.
- 7 Check one or more types of items you want to include in the compare/merge session, and click **Next**.
- 8 Check one or more compare/merge options to use at the beginning of the compare/merge.
- 9 Click **Finish**.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[VCM Merge Types, Rules, and Scenarios](#)

[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

[Merging Changes from a Child View to a Parent View \(Promote\)](#)

[Merging Changes from a Parent View to a Child View \(Rebase\)](#)

[Resolving Merge Conflicts in View Compare/Merge](#)

[Using Process Items to Merge Related Files](#)

[Testing View Compare/Merge Changes](#)

[Saving a VCM Session to a Change Package](#)

[Exporting a View Compare/Merge Session](#)

[Viewing and Printing a VCM Session Summary](#)

[Viewing and Printing a View Compare/Merge Difference Report](#)

[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Merging Changes from a Child View to a Parent View (Promote)

Promote compares or merges changes from a child, or derived, view to its parent. You can start a Promote View Compare/Merge session from either a parent view or a child view.

To perform a Promote operation starting from the parent view

- 1 Open the parent view you want to use in your View Compare/Merge session.

Note: The parent is the target in a Promote operation and its tip configuration must be used.

- 2 Choose **View** ► **Compare/Merge**, or right-click selected items in the upper pane and choose **Advanced** ► **View Compare/Merge**.

This opens the **View Compare/Merge Wizard**.

- 3 Select **Promote** as the compare/merge type.
- 4 Select **Target Of Merge** under **Use Current View (name) For**, and click **Next**.

Note: Both the **Source Of Merge** and **Target Of Merge** options can be enabled if the parent view is both the parent of one or more children and a child of another view.

A **View Tree** displays all the views in the project.

- 5 Select the child view to use as the source view for the comparison and click **Next**.
- 6 Choose the source view configuration to use: **Current**, **Labeled**, **Promotion State**, or **Configuration As Of** (a specific timestamp).
Click **Next**.
- 7 Check one or more types of items you want to include in the compare/merge session and click **Next**.
- 8 Check one or more compare/merge options to use by default at the beginning of the compare/merge, and click **Finish**.

View Compare/Merge begins comparing the views immediately and opens a new StarTeam window in the **Merge Perspective** where you can resolve any remaining merge conflicts. Items with unresolved differences are easy to find. In the **Merge Status** column for each of these items, View Compare/Merge displays a question mark, followed by the word **Unresolved**. The component tabs and folders also display a question mark if they contain items with unresolved differences.

You must resolve each of these merge conflicts to complete the merge and commit the changes. See "Resolving Merge Conflicts" in the links below.

To perform a Promote operation starting from the child view

- 1 Open the child view you want to use in your View Compare/Merge session.

Note: The **View Compare/Merge Wizard** uses the configuration to which the view is currently set if you start the wizard from the source view.

- 2 Choose **View** ► **Select Configuration** and specify the source view configuration you want to use by selecting the appropriate configuration type: **Current**, **Labeled**, **Promotion State**, or **Configuration As Of**.
- 3 Choose **View** ► **Compare/Merge**, or right-click selected items in the upper pane and choose **Advanced** ► **View Compare/Merge**.
- 4 Select **Promote** as the compare/merge type.
- 5 Select **Source Of Merge** under **Use Current View (name) For**, and click **Next**.

Note: Both the **Source Of Merge** and **Target Of Merge** options can be enabled if the child view is both a parent and a child.

- 6 Check one or more types of items you want to include in the compare/merge session, and click **Next**.
- 7 Check one or more compare/merge options to use at the beginning of the compare/merge.
- 8 Click **Finish**.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)
[Changing the View Compare/Merge Session Working Folder](#)
[Merging Changes from a Parent View to a Child View \(Rebase\)](#)
[Merging Changes Between Any Two Related Views \(Replicate\)](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Using Process Items to Merge Related Files](#)
[Testing View Compare/Merge Changes](#)
[Saving a VCM Session to a Change Package](#)
[Exporting a View Compare/Merge Session](#)
[Viewing and Printing a VCM Session Summary](#)
[Viewing and Printing a View Compare/Merge Difference Report](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Merging Changes from a Parent View to a Child View (Rebase)

Rebase compares or merges changes from a parent view to one of its child, or derived, views. You can start a Rebase View Compare/Merge session from either a parent view or a child view.

To perform a Rebase starting from the parent view

- 1 Open the parent view you want to use in your View Compare/Merge session.

Note: The **View Compare/Merge Wizard** uses the configuration to which the view is currently set if you start the wizard from the source view.

- 2 Choose **View** ► **Select Configuration** and specify the source view configuration you want to use by selecting the appropriate configuration type: **Current**, **Labeled**, **Promotion State**, or **Configuration As Of**.

Note: The **View Compare/Merge Wizard** uses the configuration to which the view is currently set if you start the wizard from the source view.

- 3 Select **Rebase** as the compare/merge type.
- 4 Select **Source Of Merge** under **Use Current View (name) For**, and click **Next**.

Note: Both the **Source Of Merge** and **Target Of Merge** options can be enabled if the parent view is both the parent of one or more children and a child of another view.

A **View Tree** displays all the views in the project.

- 5 Select the child view to use as the target view for the comparison and click **Next**.
- 6 Check one or more types of items you want to include in the compare/merge session, and click **Next**.
- 7 Check one or more compare/merge options to use at the beginning of the compare/merge.
- 8 Click **Finish**.

View Compare/Merge begins comparing the views immediately and opens a new StarTeam window in the **Merge Perspective** where you can resolve any remaining merge conflicts. Items with unresolved differences are easy to find. In the **Merge Status** column for each of these items, View Compare/Merge displays a question mark, followed by the word **Unresolved**. The component tabs and folders also display a question mark if they contain items with unresolved differences.

You must resolve each of these merge conflicts to complete the merge and commit the changes.

To perform a Rebase starting from the child view

- 1 Open the child view you want to use in your View Compare/Merge session.

Note: Use the tip revision in the child view because the child is the target in a Rebase operation.

- 2 Choose **View** ► **Compare/Merge**, or right-click the selected items in the upper pane and choose **Advanced** ► **View Compare/Merge**.
- 3 Select **Rebase** as the compare/merge type.
- 4 Select **Target Of Merge** Under **Use Current View (name) For**, and click **Next**.

Note: Both the **Source Of Merge** and **Target Of Merge** options can be enabled if the child view is both a parent and a child.

- 5 Choose the source view configuration to use: **Current**, **Labeled**, **Promotion State**, or **Configuration As Of** (a specific timestamp).
Click **Next**.
- 6 Check one or more types of items you want to include in the compare/merge session, and click **Next**.
- 7 Check one or more compare/merge options to use at the beginning of the compare/merge.
- 8 Click **Finish**.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

[Merging Changes from a Child View to a Parent View \(Promote\)](#)

[Merging Changes Between Any Two Related Views \(Replicate\)](#)

[Resolving Merge Conflicts in View Compare/Merge](#)

[Using Process Items to Merge Related Files](#)

[Testing View Compare/Merge Changes](#)

[Saving a VCM Session to a Change Package](#)

[Exporting a View Compare/Merge Session](#)

[Viewing and Printing a VCM Session Summary](#)

[Viewing and Printing a View Compare/Merge Difference Report](#)

[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Naming a Change Package

To name a change package

- 1 In the Cross-Platform Client window, open one of the views you want to use in a View/Compare Merge session. It can be either the source view or the target view. Select either the entire view, or select specific items you want to compare or merge.
- 2 Start the **View Compare/Merge** wizard from the appropriate Item's **Advanced ▶ View Compare/Merge** menu (the View, Folder, or Item menu).
- 3 **View Compare/Merge** wizard to open the VCM session containing the selected items.
- 4 You can choose to save the session before you resolve any conflicts or after you have finished and completed the session.
- 5 On the VCM Session menu, choose **VCM Session ▶ Save**.
 - ◆ **Save** opens the **Change Package Properties** dialog box where you can type in the name for your change package if you do not want to use the default name provided.

Note: Once a change package has been saved, you can only change the name of a new, uncommitted change package. You can no longer change the name of a committed change package.

- 6 To change the name of a new uncommitted change package, select it in the **Change Perspective** to open it in the VCM session. Choose **VCM Session ▶ Properties** and change the name in the **Properties** dialog box.

Related Procedures

[Saving a VCM Session to a Change Package](#)

[Exporting a View Compare/Merge Session](#)

[Editing Change Package Properties](#)

Opening a Change Package in a VCM Session

This procedure presumes you have already saved one or more change packages that are visible in **Change Perspective** of the target view in which it resides.

Note: Each change package is saved in the target view.

To open a change package in an a VCM Session

- 1 In the Cross-Platform Client window, open the target view of the change package you want to open in a VCM session.
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 In this list of change packages, select the change package you want to open.
- 4 Do one of the following:
 - ◆ If the selected change package is new and uncommitted, choose **Change Package ► Open (in VCM)**, or double-click it. Once a change package is opened in VCM, you can view and modify it, and either save it again for more work later on, or commit it.
 - ◆ If the selected change package is committed, choose **Change Package ► Open as Read Only (in VCM)** or double-click it. Once the session is opened in VCM, you can view it, but you can not make any changes.

Note: Alternatively you can double-click a change package and it will open in a VCM session. If it is committed, it will automatically open as read-only. The menu items for opening a change package in a VCM session are also on the context menu of the selected change package.

Related Concepts

[Change Packages Overview](#)
[Change Perspective UI](#)

Related Procedures

[Editing Change Package Properties](#)
[Editing a Change Package](#)
[Replaying a Change Package to Another View](#)

Opening a Change Package in Read-Only Mode for Review

This procedure presumes you have already saved one or more change packages which will be visible in **Change Perspective** of the target view in which it resides.

Note: Each VCM change package is saved in the target view.

To open a change package in read-only mode

- 1 In the Cross-Platform Client window, open the target view of the change package you want to use.
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 Select the change package you want to view.
- 4 Choose **Change Package** ► **Open as Read Only (in VCM)**.

The selected change package opens in a VCM session. You can view all the changes that were made, or view proposed changes to a yet uncommitted change package. However, you may not make any changes yourself or save the session.

Related Concepts

[Change Packages Overview](#)
[Change Perspective UI](#)

Related Procedures

[Replaying a Change Package to Another View](#)

Opening a Saved or Exported View Compare/Merge Session

You can save or export a View Compare/Merge session to a shortcut file (.vcms or .vcmx), and later restore the session for more work.

- ◆ If you save the session to a .vcms file, you can open it again on the same computer.
- ◆ If you export the session to a .vcmx file, you can move the file to a different computer and restore the session there.

Note: The .vcmx file contains more information than a .vcms file, such as merged result files.

To restore a previously saved or exported View Compare/Merge session

- 1 Open the StarTeam client.
- 2 Choose **Project** ► **Open Shortcut** from the client main menu to display the **Open** file browser.
- 3 Select **VCM Session (*.vcmx;*.vcms)** in the **Files of Type** field.
- 4 Browse to the location of the session file you want to restore and click **Open**.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

[Testing View Compare/Merge Changes](#)

[Viewing and Printing a View Compare/Merge Difference Report](#)

[Saving a VCM Session to a Change Package](#)

[Exporting a View Compare/Merge Session](#)

[Deleting a View Compare/Merge Session](#)

Preparing Your Files for a View Compare/Merge Session

Before starting a View Compare/Merge session, take the following steps to prepare your files.

Note: To compare or merge specific items in a view, or all items in specific folders in the view, start the View Compare/Merge session from the **Advanced** context menu on the selected items, rather than from the **View** menu. When you start a View Compare/Merge session from the **View** menu, all items of requested types and folders in the view are included.

To prepare your files for a view compare/merge session

- 1 Check in any modified files that will be part of the View Compare/Merge session, and optionally create a view or revision label to use for the source.
- 2 Select the views you want to compare and set their properties to permit items to branch on change.

Note: The target view is the only view that receives the results of the merge.

- 3 Decide whether you want to merge all changes in a view, or merge changes to selected items in a view. Select individual items in the upper pane if you are including specific items in the View Compare/Merge session. You can select multiple items on multiple component tabs, including the **Folder** tab.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Merging Changes from a Child View to a Parent View \(Promote\)](#)
[Merging Changes from a Parent View to a Child View \(Rebase\)](#)
[Merging Changes Between Any Two Related Views \(Replicate\)](#)
[Using Process Items to Merge Related Files](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Replaying a Change Package to Another View

The updates performed by a change package can be reapplied to another view. The **Change Perspective** in the Cross-Platform Client lists committed change packages and provides a drag-and-drop gesture for applying a selected change package to another view. The new target view must be a root, branching, non-derived, or updateable reference view. Reapplying a change package to another view is called *replay*.

In a replay, the change package's original target view becomes the source view. Replaying a change package initiates a VCM operation that selects the items updated by the change package as source items and merges them to the new target view. The replay uses a VCM merge type that depends on the relationship between the source view and new target view:

- ◆ If the new target view is the immediate parent of the source view, the replay is performed as a VCM *promote* operation.
- ◆ If the new target view is an immediate child of the source view, the replay is performed as a VCM *rebase* operation
- ◆ If the new target view is neither the immediate parent or an immediate child of the source view, the replay is performed as a VCM *replicate* operation.

To replay a change package to another view

- 1 In the Cross-Platform Client window, open the target view containing the change package you want to replay..
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 In the **Change Perspective**, depending on which layout you are using, click either the **Replay** icon on the right, or the **Replay** tab at the bottom.
- 4 In the list on the change packages, click the committed change package you want to replay.
- 5 Drag that change package to the Replay viewer and drop it on the target view to which you want those changes applied.

When a change package is replayed, a new change VCM session opens in the new target view displaying the changes to be applied. The new session acts as a staging area for the replayed updates so that they can be reviewed, tested, and adjusted. Just as with all VCM sessions, a session created by a replay can be saved, restored, and refreshed any number of times before it is committed.

- 6 Optionally, make any additional changes to the new change package, then either save the change package or commit the changes.

When a VCM Session created by a replay is committed, a new change package is generated that belongs to the new target view. As a change package is replayed to other views, and as the resulting descendant change packages are replayed to yet more views, the replays form a “tree” that can be queried in the Cross-Platform Client showing the history of how a change package has been propagated to other views.

Note: Icons in front of change packages are different in the Replay viewer are different for new, uncommitted change packages than for committed change packages. The change package icon for a New change packages contains a red arrow. The committed change package icon is the same, but without the red arrow..

Related Concepts

[Change Packages Overview](#)

[Change Perspective UI](#)

Related Procedures

[Editing a Change Package](#)

Resolving File Differences from a View Compare/Merge Session

If you have content differences between two text files, you can open File Compare/Merge directly from the View Compare/Merge session to resolve the differences. You can also change the action from **Merge** to **Overwrite**, **Mark As Resolved**, or **Ignore**.

To resolve file content differences in VCM

- 1 Start a View Compare/Merge session.
By default, the View Compare/Merge window opens in the **Merge Perspective**. Stay in this perspective.
- 2 Select a file with differences (having a merge status of either **Resolved** or **Unresolved**).
- 3 Click the **Property** tab in the lower pane.
The **<File Content>** property is displayed in the **Property** pane with boldface type since it has differences.
- 4 Right-click in the row for the **<File Content>** property and choose an action to resolve the conflict.
Choose **Use Source** to use the version of the file in the source view, **Use Target** to use the version in the target view, or **Merge** to open File Compare/Merge to compare and merge the contents.

Tip: To quickly specify a resolution, double-click the **Source** or **Target** column for the **<File Content>** property to use that value for the **Merged** column. You can also double-click the **Merged** column for the **<File Content>** property to open File Compare/Merge to merge the files.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Views](#)
[Resolving Merge Conflicts in View Compare/Merge](#)
[Merging a Local File with the Tip Revision](#)
[Saving a VCM Session to a Change Package](#)
[Generating Reports from a File Compare/Merge Session](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Resolving Merge Conflicts in View Compare/Merge

You can resolve merge conflicts in the **Merge Perspective** or the **Compare Perspective** of a View Compare/Merge session. You can also open File Compare/Merge directly from the View Compare/Merge session to resolve differences in text file contents.

Note: When merging views, no **Merge** or **Mark As Resolved** actions are possible for nonbranching items, which are requirements, tasks, and topics.

To resolve merge conflicts in a View Compare/Merge session

- 1 Choose **VCM Session** ► **Merge Perspective** or **VCM Session** ► **Compare Perspective** to switch to one of these two perspectives.
- 2 Double-click an item with an **Unresolved**, **Resolved**, or **Merge** status.

Note: The item status displays in the **Merge Action** column of the **Merge Perspective**, or in the **Action** column of the **Compare Perspective**.

The **Resolve/Merge Properties** dialog box opens displaying the item properties.

- 3 Select a property with a conflict to resolve.

Note: Properties with conflicts display in **boldface** type.

- 4 Click the **Use Source**, **Use Target**, or **Merge** button.

Tip: To quickly specify an action, double-click the **Source** or **Target** column for the property to use that value in the **Merged** column.

- 5 Optionally, double-click the **Merged** column on a text file to resolve the text differences manually.

When an unresolved item is resolved, View Compare/Merge updates the status to **Resolved** and changes the status icon to a check mark.

Tip: You can also resolve conflicts directly in the **Property** pane of the **Merge Perspective**. Right-click a property and choose an action from the context menu.

Note: If you use need to stop work on the VCM change session before resolving and committing the changes, or you want to finish it later after having someone else review the proposed changes, you can use one of the **VCM Session** menu items **Save**, **Save to Local**, or **Export** before committing the session. StarTeam creates a *Change Package* object with a unique name in the target view which contains all the changes proposed for that VCM session. It is then available in the **Change Perspective** where it can later be restarted, reviewed, committed, and once committed, have the change package replayed, which reuses the change package to apply the same set of changes to another view.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[View Compare/Merge Actions](#)

[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)

[Resolving File Differences from a View Compare/Merge Session](#)

[Testing View Compare/Merge Changes](#)

[Saving a VCM Session to a Change Package](#)

[Changing View Compare/Merge Actions](#)

[Running View Compare/Merge from the Command-line](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Restarting a VCM Session

This procedure describe how to restart an uncommitted VCM session so you can start completely over with the same initial session. There are two ways to restart a VCM session. Both result in the same outcome.

Restarting a VCM session restores it to the same state it was in when you finished running the **View Compare/Merge** wizard to originally start the session. If you previously resolved any conflicts before you restart the session, all those changes will disappear, and you will be returned to the beginning of the VCM session where you can start over reviewing and editing changes.

To restart a VCM session

- 1 In the Cross-Platform Client window, open the target view of the change package you want to use.
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 Choose the **New** (uncommitted) change package you want to restart.
- 4 Do one of the following:
 - ◆ In the **Change Perspective**, choose **Change Package** ► **Restart Session** . The selected session will open in a VCM session in its original state immediately after running the **View Compare/Merge** VCM wizard.
 - ◆ In the **Change Perspective**, choose **Change Package Open (in VCM)**, or double-click the selected session. This opens the change package in the VCM session as it was last saved with the most recent changes to the session. On the VCM session menu, choose **VCM Session** ► **Restart Session**. Now all the changes made are removed and the session returns to the state it was in right after running the **View Compare/Merge** wizard.

The advantage of the **Restart Session** operation lets you completely start over without having to re-run the **View Compare/Merge Wizard**. This can save a lot of time for especially large VCM sessions.

Related Procedures

[Resolving Merge Conflicts in View Compare/Merge](#)
[Saving a VCM Session to a Change Package](#)

Running View Compare/Merge from the Command-line

The View Compare/Merge Utility ([VCMUtility](#)) is a command-line utility that compares a source StarTeam view to a target view, and optionally merges the differences into the target view.

To run the VCMUtility

- 1 Open a command window and go to the StarTeam installation root folder.
- 2 Type [VCMUtility](#), followed by any desired options.

```
VCMUtility [<options file>] [options]
```

Tip: To see a list of all available commands and options, type [VCMUtility -help](#) in the command window.

Note: You can provide options in the specified [<options file>](#) (as the first parameter), command-line arguments, or both. Command-line arguments override any options found in the [<options file>](#). In the [<options file>](#), start option names in column 1 and exclude the leading “-”.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

[VCM Merge Types, Rules, and Scenarios](#)

[View Merge Type Scenarios](#)

Saving a VCM Session to a Change Package

There are several ways to save a VCM session: **VCM Session ▸ Save**, and **VCM Session ▸ Save to Local**. However, only **VCM Session ▸ Save** saves the session as a change package.

Note: To open a VCM session on a different computer, export the file instead of saving it by using **VCM Session ▸ Export** menu. **Export** creates a VCM exchange file (.vcmx) which allows the entire VCM session to be transported to another computer, allowing the session to be imported from the **Project ▸ Open Shortcut** menu on that computer.

To save a View Compare/Merge Session

- 1 Start a View Compare/Merge Session and resolve the conflicts in it.
- 2 Choose **VCM Session ▸ Save**.
The **Change Package Properties** dialog box opens.
- 3 Make any desired changes to the change package properties and click **Save**.
This saves the VCM Session as a change package object into the target view of the StarTeam Server.

Once you have saved your VCM session, you can continue editing your session until you are ready to save it again or commit it.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Exporting a View Compare/Merge Session](#)

[Opening a Change Package in a VCM Session](#)

[Opening a Saved or Exported View Compare/Merge Session](#)

Setting Locking Options for a View Compare/Merge Session

Locking source files prevents someone else from modifying them (in the tip configuration) while they involved in a View Compare/Merge session. Locks apply to the "whole item" within the view regardless of timestamp.

When starting a View Compare/Merge session, you can enforce default locking behavior by setting one or more locking options in the **View Compare/Merge Wizard**. Once in the session, you can override the lock status of any items.

When you close a View Compare/Merge session, you are prompted as to whether you want to retain the locks created by the View Compare/Merge session.

To set locking options for a View Compare/Merge Session

- 1 Open the source or target view to use for the View Compare/Merge session.
- 2 Choose **View** ► **Compare/Merge**.
- 3 On the **Set Options** page of the **View Compare/Merge Wizard**, check the locking options you want to enforce in the View Compare/Merge session.

Tip: To view more lock-related fields in a View Compare/Merge session, right-click a column header and choose **Show** ► **Fields** to open the Show Fields dialog box where you can select additional field columns to add to the upper pane.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing and Merging Views](#)
[Comparing Selected Items in the Source and Target View](#)
[Finding and Breaking Locks on Items with Differences](#)
[Changing View Compare/Merge Actions](#)
[Viewing and Printing a VCM Session Summary](#)
[Viewing and Printing a View Compare/Merge Difference Report](#)
[Exporting a View Compare/Merge Session](#)
[Deleting a View Compare/Merge Session](#)
[Opening a Saved or Exported View Compare/Merge Session](#)

Related Reference

[View Compare/Merge Wizard: Set Options](#)

Specifying the Initial Perspective for a View Compare/Merge Session

To specify a startup perspective

- 1 Open one of the views you want to use in your View Compare/Merge session.
- 2 Start the **View Compare/Merge Wizard**.
- 3 On the **Set Options** page of the wizard, check **Start With Merge Perspective Current** to begin in the **Merge Perspective** (default), or uncheck it to begin in the **Compare Perspective**.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing Selected Items in the Source and Target View](#)

Related Reference

[View Compare/Merge Wizard](#)

Testing View Compare/Merge Changes

To test proposed changes in a View Compare/Merge session

- 1 Verify that you have resolved all merge conflicts in the **Compare Perspective** or **Merge Perspective** of the View Compare/Merge session.

Note: You cannot check out individual files with merge conflicts in the **Test Perspective** of a View Compare/Merge session. Resolve all conflicts before using the **Test Perspective**.

- 2 Choose **VCM Session** ► **Test Perspective** and check out the files in your project, including those with merged changes.

View Compare/Merge saves all files locally to a temporary default working folder until you commit the View Compare/Merge session.

Tip: You can specify an alternate View Compare/Merge working folder. Choose **VCM Session** ► **Working Folder** in the **Test Perspective**.

- 3 Use your build tools to test that the project will build, run, and has the needed changes.

Note: You cannot build or run from within StarTeam.

- 4 Optionally, switch to the **Merge Perspective** or the **Compare Perspective** and make changes as needed.
- 5 Recheck out your project if you have made additional changes, and rerun your tests.
- 6 Choose **VCM Session** ► **Commit Changes** when you are satisfied with all the proposed changes.

View Compare/Merge applies all the changes to the StarTeam server repository and closes the session.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)

Related Procedures

[Comparing and Merging Views](#)
[Changing the View Compare/Merge Session Working Folder](#)
[Comparing and Merging Views](#)
[Changing View Compare/Merge Actions](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)
[Conditions for VCM Action Decisions](#)

Using Filters and Queries to Navigate Change Packages

The Cross-Platform Client allows users to list all active and committed change packages for a given view in the **Change Perspective**. The change package list can be sorted and filtered by various criteria. A change package can be selected, and certain details can be displayed. Only base properties such as the name, description, and lock state of uncommitted change packages can be viewed in the list. For committed change packages, details of each updated item can be viewed such as its properties and history. Other information about the change package, such as VCM options used, can also be displayed. More details on interaction with change packages in the CPC are described later.

This procedure presumes you have already saved one or more change packages, which are visible in **Change Perspective** for the target view of interest. Each VCM change package from the VCM session is saved in the target view.

To use filters and queries in the Change Perspective

- 1 In the Cross-Platform Client window, open the target view containing the change package you want to use.
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 Choose **Change Package** ► **Filters** to open the standard Cross-Platform Client **Filter** menu.
- 4 Create, modify, and select the various filters and queries you want to use for change packages in the **Change Perspective**. Use exactly the same processes you use to create filters and queries in the Cross-Platform Client.
- 5 When you are finished, save any new or modified filters and queries. The new filters will display in the **Filter** drop-down list above the list of change sessions.

Related Concepts

[Change Packages Overview](#)
[Change Perspective UI](#)

Related Procedures

[Creating Filters](#)
[Copying Filters](#)
[Editing Filters](#)
[Creating Queries](#)

Using Process Items to Merge Related Files

View Compare/Merge allows you to compare and merge one or more sets of files related to specific process items by selecting the process items to which they are linked and pinned (instead of selecting the files themselves).

To compare and merge files based on their process items

- 1 Choose **Project** ► **Open** on the StarTeam client menu, and open the source view containing the process items you want to use.
- 2 Choose **View** ► **Select Configuration** and specify the source view configuration you want to use by selecting the appropriate configuration type: **Current**, **Labeled**, **Promotion State**, or **Configuration As Of**.

Note: The **View Compare/Merge Wizard** uses the configuration to which the view is currently set if you start the wizard from the source view.

- 3 Select the process items from the **Change Request**, **Requirement**, or **Task** tabs in the source view.

Note: Use **SHIFT+CLICK** or **CTRL+CLICK** to select multiple items in the upper pane. The items you select stay selected when you switch tabs or project windows.

- 4 Right-click any of the selected process items in the source view and choose **Advanced** ► **View Compare/Merge**.

This opens the **View Compare/Merge Wizard**.

- 5 Select the type of compare or merge to perform: **Promote**, **Rebase**, or **Replicate**.
- 6 Select **Source Of Merge** under **Use Current View [name] For**, and click **Next**.
- 7 Check **Changes associated with [item]** on the **Include Selected Items** page of the wizard.
- 8 Check **Selected [item]** to compare and merge the selected items as well as the files linked to them.

Warning: The link to each file must be pinned at both ends or it is not included.

- 9 Click **Next** and complete the wizard.

View Compare/Merge begins comparing the views immediately and opens a VCM session window in the **Merge Perspective** where you can resolve differences in the process items, and/or the files linked and pinned to them. Items with unresolved content or property differences are easy to find. In the **Merge Status** column for each of these items, View Compare/Merge displays a question mark, followed by the word **Unresolved**. The component tabs and folders also display a question mark if they contain items with unresolved differences.

If View Compare/Merge can propose an appropriate merge action, it displays the item's status as **Resolved** with a green check mark. You can change any proposed action with another possible action.

Note: You must resolve each of these merge conflicts before you can complete the merge by committing the changes.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)

[Testing View Compare/Merge Changes](#)

[Saving a VCM Session to a Change Package](#)

[Exporting a View Compare/Merge Session](#)

[Viewing and Printing a VCM Session Summary](#)

[Viewing and Printing a View Compare/Merge Difference Report](#)

[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Viewing and Printing a VCM Session Summary

At any time during a View Compare/Merge session, you can view the session summary.

To view the VCM session summary

- 1 Start a View Compare/Merge session.
- 2 Choose **VCM Session** ► **Summary**.
The **VCM Session Summary** dialog box opens and displays a two-column list of the session properties.
- 3 Optionally, click the **Browse** button to choose a different path and name for the output filename.
The report is saved as an `.html` file to the specified location.
- 4 Optionally, click **View In Browser** to display and/or print the session summary in your default browser.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)

[Comparing and Merging Views](#)

[Changing the View Compare/Merge Session Working Folder](#)

[Testing View Compare/Merge Changes](#)

[Saving a VCM Session to a Change Package](#)

[Viewing and Printing a View Compare/Merge Difference Report](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Viewing and Printing a View Compare/Merge Difference Report

At any time during a View Compare/Merge session, you can view a report of the differences in the session to see what you have done and what you have left to do. There are two types of difference reports available:

- ◆ **Difference Report:** Displays information about the VCM session parameters and options, a summary of the number of differences by type and action, and the view differences between the source and target by file and folder.
- ◆ **Detailed Difference Report:** Displays the same information as the **Difference Report**, but also includes information about items whose parent folder has moved in the source or target view. If no folders were moved, this report will look the same as the **Difference Report**.

To view and print a Difference Report for a View Compare/Merge session

- 1 Start a View Compare/Merge session, or open a saved change package from the **Change Perspective** using **Change Package ► Open (in VCM)**, or **Change Package ► Open as Read Only (in VCM)**.
- 2 Choose **VCM Session ► Difference Report** or **VCM Session ► Detailed Difference Report**.
The **VCM Session Differences** dialog box displays showing a two-column list of the session differences.
- 3 Optionally, click the **Browse** button to choose a different path and name for the output filename.
The report is saved as an `.html` file.
- 4 Press the **View In Browser** button to display the difference report in your default browser.
- 5 Use the default browser menus to print the report.

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing and Merging Views](#)
[Changing the View Compare/Merge Session Working Folder](#)
[Testing View Compare/Merge Changes](#)
[Saving a VCM Session to a Change Package](#)
[Exporting a View Compare/Merge Session](#)
[Viewing and Printing a VCM Session Summary](#)

Related Reference

[Conditions for VCM Action Decisions](#)

Viewing Change Package Details, History and Labels

This procedure presumes you have already saved one or more change package sessions which will be visible in **Change Perspective** for the target view of interest. Each VCM change package from the VCM session is saved in the target view.

To view details, history, or label information for a change package

- 1 In the Cross-Platform Client window, open the target view of the VCM change session you want to use.
- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 Choose one of the viewers you want to see from the icons at the right in the “side by side” layout, or from the tabs at the bottom of the “top over bottom” layout.
 - ◆ The **Details** pane displays information specific to the selected change package, such as it's description, by whom it was created, when it was last modified, if it was committed, when it was committed and by whom. It also displays the session type (promote, rebase, or replicate), source and target view, transaction ID, and labels. The **Details** pane can be customized like the **Details** pane for other StarTeam items. Sample Detail pane templates are installed under the Cross-Platform Client root installation folder in the [samples\details-templates](#) folder. There is a [changepackage.details.html](#) which you can use to modify the fields displayed on the **Details** pane in the **Change Perspective**.
 - ◆ The **History** pane displays all the historical versions of the selected change package, complete with Dot Notation, who modified each revision, and the description of the revision. As with the **History** pane for other StarTeam items, there is a context menu for performing standard History actions, and right-clicking on the fields at the top let's you change which fields are displayed and the order in which they are displayed.
 - ◆ The **Label** pane in the **Replay** view works exactly the same as for other StarTeam items. It shows all the branch revisions and their date/time information, and you have the same context menu as with other items to perform the same label functions.

Related Procedures

[Customizing the Detail Pane](#)

Viewing Changes in a Change Package

This procedure presumes you have already saved and committed one or more change packages, which will be visible in **Change Perspective** of the target view in which it resides.

Note: Each change package from a VCM session is saved in the target view.

To view the changes from a change package

- 1 In the Cross-Platform Client window, open the target view containing the change package you want to view.

Note: Unless you have a view opened, there is no way to view the **Change Perspective**.

- 2 On the right side of the toolbar, click the **Show Change Perspective** icon to switch to the **Change Perspective**.
- 3 Select the change package you want to view in the list on the left.
- 4 Click the **Show Change Viewer** icon in the “side by side” layout, or click the **Change** tab in the “to over bottom” layout.

This displays the details of every change made in the change package, including among other things, each item type changed, the change type used, and the action used to resolve the conflict or difference.

Alternatively, you can open the change package as a VCM session, but it will be in Read-Only mode since the changes have already been committed.

Note: You can also view the changes in a new (uncommitted session), but you must open a new change package as a VCM session to see the proposed changes and resolutions.

Viewing the Change Perspective

The Cross-Platform Client now has two perspectives available in the main window: The standard **Content Perspective** and the **Change Perspective**, the latter of which lets you view the details of all the changes made in committed View Compare/Merge (VCM) sessions, or changes proposed in uncommitted VCM change sessions. The **Change Perspective** also lets you apply the same changes made in one committed session to another project or view (*Replay*). This is possible because all the changes for a single VCM session are recorded in a *Change Package* object created by the VCM session when it is saved or committed. The **Content Perspective** also allows you to manage other aspects of a change package as you can with other StarTeam objects.

The Cross-Platform Client automatically opens in the **Content Perspective**.

To switch to the Change Perspective

- 1 In the Cross-Platform Client window, click the **Show Change Perspective** icon at the right of the toolbar.
- 2 To switch back to the **Content Perspective**, click the **Show Content Perspective** icon at the right of the toolbar.

Comparing and Merging Files and Folders

This section contains procedures for comparing and merging files, folders, and images. How you choose to use File Compare/Merge depends on what type of files you are comparing and/or merging.

To compare a local file with a revision in the StarTeam repository, or to compare two historical revisions, open the main File Compare/Merge from inside the StarTeam client, or display the embedded File Compare/Merge.

To compare two local files, open the standalone File Compare/Merge from the **Start** menu.

In This Section

[Applying Character Encoding to Files](#)

Describes how to apply a character encoding set to a file in File Compare/Merge.

[Comparing a Local File with a Repository File](#)

Describes how to use File Compare/Merge to compare a local file with a revision of the file in the repository.

[Comparing and Merging Files](#)

Describes the core process for performing a file compare and merge operation.

[Comparing Folders](#)

Describes how to compare two folders using File Compare/Merge.

[Comparing Historical File Contents](#)

Describes how to compare the contents of two historical files in the repository.

[Comparing Images](#)

Describes how to compare two images using File Compare/Merge.

[Comparing Two Local Files](#)

Describes how to compare local files using File Compare/Merge.

[Editing Files in a File Compare/Merge Session](#)

Describes how to edit text files in a File Compare/Merge session.

[Generating Reports from a File Compare/Merge Session](#)

Describes the procedures for generating and printing reports from a File Compare/Merge.

[Merging a Local File with the Tip Revision](#)

Describes how to use File Compare/Merge to merge the contents of a local file with the tip revision of the file in the StarTeam repository.

[Merging Folders](#)

Describes how to merge the contents of folders.

[Merging Two Local Files](#)

Describes how to use File Compare/Merge to merge the contents of two local files.

[Saving Files Modified in a File Compare/Merge Session](#)

Describes how to save files modified during a File Compare/Merge session.

[Setting File Compare/Merge Options](#)

Describes how to customize the display and behavior of File Compare/Merge.

Applying Character Encoding to Files

By default, when you open a file in File Compare/Merge, it uses the system-dependent default encoding for the loaded file. The file character set encoding is displayed on the **Status Bar**, in two or three boxes (one for each pane). Initially, these boxes display the text **Enc:Default** when the files are opened. You can change this encoding for each file in the File Compare/Merge panes.

To apply a character encoding to a file

- 1 Click the Character set encoding box on the File Compare/Merge **Status Bar** that corresponds to the pane containing the file you want to change.

This opens the **Charset Selection** dialog box which displays a list of character sets.

- 2 Select the desired character set from the list.
If you selected a valid character set, the file immediately switches to that character set.
- 3 Close the **Charset Selection** dialog box.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)

[Comparing a Local File with a Repository File](#)

[Comparing Historical File Contents](#)

[Comparing Two Local Files](#)

[Editing Files in a File Compare/Merge Session](#)

[Merging a Local File with the Tip Revision](#)

Comparing a Local File with a Repository File

To compare a local file with a revision in the repository

- 1 Select a StarTeam repository file to compare with your local file.

Note: You can select the tip revision a file in the upper pane, or a previous revision on the **History** tab.

- 2 Right-click the selected file and choose **Compare Contents**.
The File Compare/Merge session opens in a separate window, with each file in its own edit pane.
- 3 Click the DOWN ARROW to the right of the **Switch Comparison Modes** toolbar button and select a comparison mode.
- 4 Perform any required edits to the files directly in the edit panes.

Tip: You can also use the embedded **Compare** panes to quickly compare two files. Selecting two files in StarTeam, and choose **Tools ▸ Compare** . You can view files but you cannot edit files in the embedded **Compare** panes.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Historical File Contents](#)
[Comparing Folders](#)
[Comparing Images](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)
[Generating Reports from a File Compare/Merge Session](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Actions](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Comparing and Merging Files

This procedure presents the basic high-level tasks involved in comparing and merging a local file with a file in the StarTeam repository. You can find additional procedures in Comparing and Merging Files and Folders in the Procedures section of the Contents pane in the online help.

To perform a file compare/merge

- 1 Start a File Compare/Merge session to compare two files.
[Comparing a Local File with a Repository File](#)
- 2 Resolve any differences between the files in the File Compare/Merge window.
[Editing Files in a File Compare/Merge Session](#)
- 3 Generate a report of the changes made during the File Compare/Merge session.
[Generating Reports from a File Compare/Merge Session](#)
- 4 Save the merge changes.
[Saving Files Modified in a File Compare/Merge Session](#)

Note: You can also merge the contents of the two files.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Merging a Local File with the Tip Revision](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Merging Two Local Files](#)
[Comparing Folders](#)
[Comparing Images](#)
[Merging Folders](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Comparing a Local File with a Repository File

To compare a local file with a revision in the repository

- 1 Select a StarTeam repository file to compare with your local file.

Note: You can select the tip revision a file in the upper pane, or a previous revision on the **History** tab.

- 2 Right-click the selected file and choose **Compare Contents**.
The File Compare/Merge session opens in a separate window, with each file in its own edit pane.
- 3 Click the DOWN ARROW to the right of the **Switch Comparison Modes** toolbar button and select a comparison mode.
- 4 Perform any required edits to the files directly in the edit panes.

Tip: You can also use the embedded **Compare** panes to quickly compare two files. Selecting two files in StarTeam, and choose **Tools ▸ Compare** . You can view files but you cannot edit files in the embedded **Compare** panes.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Historical File Contents](#)
[Comparing Folders](#)
[Comparing Images](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)
[Generating Reports from a File Compare/Merge Session](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Actions](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Editing Files in a File Compare/Merge Session

To edit a file in a File Compare/Merge pane

- 1 Start a main or standalone File Compare/Merge session on two files you want to compare.
The File Compare/Merge panes display the differences between the two files in color differences blocks.
- 2 Edit the files by modifying text directly in the edit panes.
Use standard editing techniques or use the dynamic editing buttons in the color differences blocks.

Note: You cannot modify the actual revision of a file in the StarTeam repository. If one of the files you are comparing is a revision in the repository, File Compare/Merge creates a temporary copy of the file locally, which you can also edit and save locally with a different filename.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Comparing Folders](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)
[Comparing Images](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Generating Reports from a File Compare/Merge Session

This procedure describes how to generate a difference report from File Compare/Merge session. It assumes that you are in a compare/merge session and want to generate a report of your changes before you commit them and close the session.

To generate a report from File Compare/Merge

- 1 Right-click in a File Compare/Merge pane and choose **Generate Report** to open the **Report** dialog box.

Note: If you are in the main or standalone File Compare/Merge window, you can use the **Generate Report** toolbar button.

- 2 Choose a report generator (transformer) from the drop-down list in the **Report** dialog box.
- 3 Optionally, when available, modify the report parameters to customize the report appearance.
- 4 Click **Next** to preview the report.
- 5 Click **Save** and browse to the target location for the report.
Choose a file extension that is valid for the output type (format) of the transformer you selected, and click **Save**.
- 6 Click **Close** in the **Report** dialog box.

Note: To print your report, open it in an appropriate application such as a text editor or a browser, and print.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)
[Setting File Compare/Merge Options](#)
[Comparing a Local File with a Repository File](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Generating Reports from a File Compare/Merge Session](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)

Saving Files Modified in a File Compare/Merge Session

If you edit a file during a File Compare/Merge session, you can save the edited file from the File Compare/Merge window, or be prompted to save the edits when you exit File Compare/Merge.

To save a modified file in File Compare/Merge

- 1 Start File Compare/Merge and make your edits and/or merge the files.

The **Save Changed Text File** button becomes active above each pane containing modifications. It is located to the right of the **File Path/Name** field directly above the editing pane.

- 2 Save the modified files using one of the methods below:

- ◆ Click the **Save Changed Text File** button above each pane that contains the edits you want to save. The file is saved with the existing name and path, which is displayed in the **File Path/Name** field.
- ◆ Click the DOWN ARROW beside the **Save Changed Text File** button and choose **Save As** to save the file with a different name and/or path than that displayed in the **File Path/Name** field. A file browser opens for selecting the path and specifying the filename.
- ◆ Choose **File** ▸ **Save All** to save all files that have been modified in File Compare/Merge with their current names and paths.
- ◆ Choose **File** ▸ **Close Current Pane** to close the current tab pane containing a File Compare/Merge session. The **Save Results** dialog box opens and prompts you to save the files that have changed. Check the specific files you want to save and click **Yes**. The files are saved with their current name and path.
- ◆ Choose **File** ▸ **Exit** to close all the active File Compare/Merge sessions. The **Save Results** dialog box opens and prompts you to save all files that have changed. Check the specific files you want to save and click **Yes**. The files are saved with their current name and path.

Note: If you edit the temporary copy of a file revision from the StarTeam repository, unless you explicitly save that file to a different location with a new name, File Compare/Merge assigns the file a new name and saves it in a **temp** folder in your local home folder. The naming convention is the word **star**, followed by numbers (for example, **star23276.xml**). The file extension remains the same as the original file.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing a Local File with a Repository File](#)

[Comparing Historical File Contents](#)

[Comparing Two Local Files](#)

[Comparing Folders](#)

[Merging a Local File with the Tip Revision](#)

[Merging Folders](#)

[Comparing Images](#)

[Editing Files in a File Compare/Merge Session](#)

[Generating Reports from a File Compare/Merge Session](#)

[Customizing Compare and Merge Reports](#)

[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)

[File Compare/Merge Keyboard Shortcuts](#)

[File Compare/Merge Actions](#)

Comparing Folders

Using the standalone File Compare/Merge, you can compare and restructure folders on your computer. You can also start a file comparison from a folder comparison. If you have not modified the **General** options settings, File Compare/Merge automatically begins comparing folders as soon as you select them.

File Compare/Merge provides a context menu and toolbar with the most commonly used actions that you will need to perform in the editor panes. These are explained in detail in the "Table of File Compare/Merge Actions" in the reference links below.

To compare two folders

- 1 From the Windows desktop, choose **Start ▶ Program Files**, navigate to your StarTeam program installation group, and choose **File Compare Merge**.
- 2 Choose **File ▶ New Folder Comparison** (CTRL+SHIFT+N) to open a new folder comparison tab.

Note: Using the **New Folder Comparison** command opens the folder comparison results with **Auto Commit On**. To open a folder comparison with **Auto Commit Off**, choose **File ▶ New Folder Merge**. Once File Compare/Merge is opened, you can click **Auto Commit Mode** on the toolbar to toggle between **Auto Commit On** or **Auto Commit Off**.

- 3 Click the **Open New Comparison** button at the top right of each editor pane, and browse to the folder you want to compare.

Note: Alternatively, you can type the path to each folder to the left of the **Save Changed Text File** button, or click the drop-down arrow on the **Open New Comparison** button and select a folder from the history list.

By default, File Compare/Merge immediately compares the selected folders, and the results display in the two panes of the File Compare/Merge window. You can change this and other default settings in the File Compare/Merge Options dialog box on the **General** and **Folder Comparison** pages.

- 4 Optionally, right-click and choose **Switch Comparison Modes** to choose a different comparison mode display. If you choose one of the **3-Way** comparison modes, you will have a third pane.
- 5 Perform any required changes to the folders directly in the edit panes. You can move, delete, and copy folders from one pane to the other, and are prompted to commit your changes.

If there are differences between individual files in the folders you are comparing, you can also start a file comparison from the folder comparison pane. Double-click the file in the folder comparison pane, or selecting it and click **Open New Comparison** on the toolbar. A new file comparison pane opens displaying the contents of the two files where you can edit, merge, and save your changes.

File Compare/Merge highlights the differences between the two folders using colors and linked lines, with the linked lines connecting the related differences between the panes.

Tip: You can open additional folder comparison windows by choosing **File ▶ New Folder Comparison** on the File Compare/Merge main menu, or by pressing CTRL+N.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Images](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Comparing Historical File Contents

You can only compare historical file contents using the main or embedded File Compare/Merge in the StarTeam client. You cannot edit historical file revisions.

To compare the contents of two historical revisions in the repository

- 1 Click the **History** tab and select two revisions to compare.
- 2 Choose **Tools** ► **Compare**.

The embedded compare panes display the contents of the two files at the bottom of the StarTeam window.

Note: You can also compare the properties of two non-file items, such as a change request, or two revisions of the same non-file item. Select the two non-file items, then choose **Tools** ► **Compare**. This opens the embedded double-paned compare view at the bottom of the window and displays the properties of the two selected items.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Two Local Files](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)

Comparing Images

You can use the standalone File Compare/Merge to compare two images on your computer. You cannot edit the files directly, but you can swap the contents of one image with the contents of another one.

File Compare/Merge provides a toolbar with a few common actions that control the editor pane display. It also includes some image comparison tools for zooming in and out, and moving the file around.

To compare two images

- 1 From the Windows desktop, choose **Start ▶ Program Files**, navigate to your StarTeam program installation group, and choose **File Compare Merge**.
- 2 Choose **File ▶ New Image Comparison** (CTRL+ALT+N) to open a new image comparison tab.
- 3 Click the **Open New Comparison** button at the top right of each editor pane, and browse to the images you want to compare.

Note: The **DOWN ARROW** to the right of the **Open New Comparison** button stores a history of your previously-opened images. You can open the **Save File As** dialog box by clicking the **DOWN ARROW** to the right of the **Save Changed Text File** button.

- 4 (Optional) Click the **DOWN ARROW** to the right of the the **Switch Comparison Modes** toolbar button and choose either a **2-Way Vertical** or **2-Way Horizontal** display.
- 5 (Optional) Click **Zoom In**, **Zoom Out**, or **Move** on the toolbar to examine your image in more detail.
- 6 (Optional) Use the double left or right-arrow buttons at the top of the image comparison panes to replace one image with the other.

The content of the source image replaces the contents of the target image without change the filename.

- 7 Click the **Save Changed Image** button at the top of the pane that contains the changed image, or close the image comparison pane and you will be prompted to save your file if you have modified it.

Tip: You can open additional image comparison windows by choosing **File ▶ New Image Comparison** on the File Compare/Merge main menu, or by pressing CTRL+ALT+N.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Folders](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Comparing Two Local Files

To compare local files using the standalone File Compare/Merge

- 1 From the Windows desktop, choose **Start ▶ Program Files**, navigate to your StarTeam program installation group, and choose **File Compare Merge**.
- 2 Click the **Open New Comparison** button at the top right of each editor pane, and browse to the files you want to compare.

Note: The **Open New Comparison** button displays a list of your previously opened files.

- 3 Right-click in any pane and choose **Switch Comparison Modes** to specify a comparison display layout.

Tip: You can open additional file comparison windows by pressing **CTRL+N**.

- 4 Edit the files by typing directly in the edit panes, or by using the dynamic editing buttons in the color difference blocks.
- 5 Optionally, right-click in one of the panes and choose **Generate Report**.
- 6 Choose **File ▶ Print** to print the report.

Note: If you want a report of your changes, you must generate the report before closing the **File Compare/Merge** window.

- 7 Choose **File ▶ Save All** to save all changes, or click the **Save Changed Text File** button above each pane to save each changed file individually.
- 8 Optionally, test your changes to ensure there are no problems, then check the change files into StarTeam.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Editing Files in a File Compare/Merge Session](#)
[Generating Reports from a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Images](#)
[Merging a Local File with the Tip Revision](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Editing Files in a File Compare/Merge Session

To edit a file in a File Compare/Merge pane

- 1 Start a main or standalone File Compare/Merge session on two files you want to compare.
The File Compare/Merge panes display the differences between the two files in color differences blocks.
- 2 Edit the files by modifying text directly in the edit panes.
Use standard editing techniques or use the dynamic editing buttons in the color differences blocks.

Note: You cannot modify the actual revision of a file in the StarTeam repository. If one of the files you are comparing is a revision in the repository, File Compare/Merge creates a temporary copy of the file locally, which you can also edit and save locally with a different filename.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Comparing Folders](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)
[Comparing Images](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Generating Reports from a File Compare/Merge Session

This procedure describes how to generate a difference report from File Compare/Merge session. It assumes that you are in a compare/merge session and want to generate a report of your changes before you commit them and close the session.

To generate a report from File Compare/Merge

- 1 Right-click in a File Compare/Merge pane and choose **Generate Report** to open the **Report** dialog box.

Note: If you are in the main or standalone File Compare/Merge window, you can use the **Generate Report** toolbar button.

- 2 Choose a report generator (transformer) from the drop-down list in the **Report** dialog box.
- 3 Optionally, when available, modify the report parameters to customize the report appearance.
- 4 Click **Next** to preview the report.
- 5 Click **Save** and browse to the target location for the report.
Choose a file extension that is valid for the output type (format) of the transformer you selected, and click **Save**.
- 6 Click **Close** in the **Report** dialog box.

Note: To print your report, open it in an appropriate application such as a text editor or a browser, and print.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)
[Setting File Compare/Merge Options](#)
[Comparing a Local File with a Repository File](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Generating Reports from a File Compare/Merge Session](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)

Merging a Local File with the Tip Revision

You can use File Compare/Merge to merge the contents of a local working file with the tip revision of the file in StarTeam. For example, if the file in your working folder has been modified, but is not based on the tip (latest) revision of this file, it is marked with a **Merge** status. This status usually occurs when you and another person have both been working on a file, but the other person has checked it in before you. To check in your edits without losing the edits of the other person, you can compare the contents of the files, reconcile the differences between them, and merge the contents into one file.

To merge a local file with the tip revision

- 1 Select a file in StarTeam with a file status of **Merge** or **Unknown**, and choose **File ▸ Check Out**.

A message appears giving you the option to merge the contents of the two files.

- 2 Click **Yes** to merge the contents.

The File Compare/Merge window opens in a **3-way Vertical** comparison mode. The two files you are comparing display in the right and left panes, and a center pane displays the merged base file. If you want to use a different comparison layout, right-click in one of the panes and choose **Switch Comparison Modes**. You can choose either **3-way Horizontal** or **3-way Mixed** comparison mode. In the **3-way Mixed** mode, the base file is in the lower pane.

Note: You can edit in all the panes in the standalone File Compare/Merge, and you can replace or append text in the base file with highlighted text from the left and right panes.

- 3 Edit and/or merge the contents of the files using one or more of the following methods:

- ◆ Perform an automatic merge: Right-click in a File Compare/Merge pane and choose **Nonconflicting Parts Auto Merge** or **Full Auto Merge**. **Nonconflicting Parts Auto Merge** places only the nonconflicting portions of the two compared files into the base file. **Full Auto Merge** parses the two files, examines the changes, and automatically decides whether to keep or add each change to the final base version of the file.
- ◆ Edit any of the files directly. You can perform common text-editing actions such as type, cut, copy, paste, and undo.
- ◆ Copy text from the right and left panes using buttons that appear in the top right corner of the color differences blocks. These buttons appear dynamically when you hover the mouse cursor over a color differences block. If you do not select specific text to copy to the center pane, File Compare/Merge copies the entire highlighted block of text.

Note: You can also use the toolbar and the context menu to navigate in the panes, control scrolling behavior, search for text, and perform other basic actions.

- 4 Optionally, right-click in one of the panes and choose **Generate Report**.

To print the report, choose **File ▸ Print**.

Note: If you want a report of your changes, you must generate it before closing the File Compare/Merge window.

- 5 Choose **File ▸ Exit** when you are satisfied with the merged results.

File Compare/Merge closes, and updates the file status to **Modified** in StarTeam.

- 6 Optionally, test the modified file to ensure there are no problems with it.
- 7 Check in the file.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Folders](#)
[Comparing Images](#)
[Merging Folders](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Merging Folders

Merging folders is quite similar to merging files, except you are merging folder contents instead of file contents. However, if two files in a folder have differences, you can open a file comparison on those files from the folder comparison pane.

You can only merge folders using the standalone File Compare/Merge.

To merge two folders

- 1 From the Windows desktop, choose **Start ▶ Program Files**, navigate to your StarTeam program installation group, and choose **File Compare Merge**.
- 2 Choose **File ▶ New Folder Comparison** (CTRL+SHIFT+N), or **File ▶ New Folder Merge** (CTRL+SHIFT+M) to open a new folder comparison tab.

Note: Using the **New Folder Merge** command opens the folder comparison results with **Auto Commit Off**. To open a folder comparison with **Auto Commit On**, choose **File ▶ New Folder Comparison**. Once File Compare/Merge is opened, you can click **Auto Commit Mode** on the toolbar to toggle between **Auto Commit On** or **Auto Commit Off**.

- 3 Click the **Open New Comparison** buttons above each editor pane, and browse to the folders you want to compare.

Alternatively, you can click the **DOWN ARROW** to the right of the **Open New Comparison** button and select a folder from the history list, or you can type the path for each folder in the directory to the left.

By default, File Compare/Merge immediately compares the selected folders, and the results display in the two panes of the File Compare/Merge window. You can change this and other default settings in the File Compare/Merge Options dialog box on the **General** and **Folder Comparison** pages.

- 4 Optionally, right-click and choose **Switch Comparison Modes** to choose a different comparison mode display. If you choose one of the **3-Way** comparison modes, you will have a third pane for viewing and editing the merged base folder.

- 5 Perform any required changes to the folders directly in the edit panes. You can move, delete, and copy folders from one pane to the other, and are prompted to commit your changes.

The toolbar and context menus are different for folder compare/merge than for file compare/merge. For folder comparisons, you have items for moving or copying files and folders left or right, deleting files and folders from the right or left panes, and several options for expanding/collapsing tree nodes.

- 6 Optionally, double-click a file with differences to open a new File Comparison pane and resolve the differences.
- 7 Optionally, right-click in one of the panes and choose **Generate Report**.

To print the report, choose **File ▶ Print**.

Note: If you want a report of your changes, you must generate it before closing the File Compare/Merge window.

- 8 Click **Save Changes To Disk** (check mark) on the toolbar to save your changes when you are finished.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Folders](#)
[Comparing Images](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Merging a Local File with the Tip Revision](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)

Related Reference

[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Merging Two Local Files

Use the standalone File Compare/Merge to compare and merge the contents of two files on your computer.

To merge two local files

- 1 Click the **Start** button on the Windows desktop and choose **Program Files ▶ Borland StarTeam ▶ StarTeam Cross-Platform Client 2009 ▶ File Compare Merge**.

- 2 Click the **Open New Comparison** button at the top right of each editor pane.

Browse to the files you want to compare, or enter the path for each file in the directory field above each pane.

Tip: The **DOWN ARROW** to the right of the **Open New Comparison** button above each pane stores a history list of the previously-opened files in that pane from which you can choose.

By default, the files are immediately compared, and displayed in the panes with line numbers.

File Compare/Merge highlights the differences between the two files using colors and linked lines, with the linked lines connecting the related differences in the two files. You can set the colors used in the editor panes in the File Compare/Merge Options dialog box on the **Color Preferences** page for **File Comparisons**.

- 3 Right-click in any pane and choose **Switch Comparison Modes**.

Choose **3-Way Vertical**, **3-Way Horizontal**, or **3-Way Mixed**.

In the **3-Way Vertical** and **3-Way Horizontal** comparison modes, the merged base file is in the center pane.

In the **3-Way Mixed** comparison mode, File Compare/Merge places the merged base file in the lower pane.

Tip: You can open additional file comparison windows by choosing **File ▶ New Comparison** on the File Compare/Merge main menu, or by pressing **CTRL+N**.

- 4 Perform any required edits to the files directly in the edit panes.

File Compare/Merge provides a context menu and toolbar with the most commonly used actions that you will need to perform in the editor panes.

- 5 Optionally, right-click in one of the panes and choose **Generate Report**.

To print the report, choose **File ▶ Print**.

Note: If you want a report of your changes, you must generate it before closing the File Compare/Merge session.

- 6 Choose **File ▶ Save All** to save all of your changes, or click **Save Changed Text File** button above each pane to save them individually.

- 7 Optionally, test the modified file to ensure there are no problems with it.

- 8 Check in the file.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Folders](#)
[Comparing Images](#)
[Merging Folders](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Saving Files Modified in a File Compare/Merge Session

If you edit a file during a File Compare/Merge session, you can save the edited file from the File Compare/Merge window, or be prompted to save the edits when you exit File Compare/Merge.

To save a modified file in File Compare/Merge

- 1 Start File Compare/Merge and make your edits and/or merge the files.

The **Save Changed Text File** button becomes active above each pane containing modifications. It is located to the right of the **File Path/Name** field directly above the editing pane.

- 2 Save the modified files using one of the methods below:

- ◆ Click the **Save Changed Text File** button above each pane that contains the edits you want to save. The file is saved with the existing name and path, which is displayed in the **File Path/Name** field.
- ◆ Click the DOWN ARROW beside the **Save Changed Text File** button and choose **Save As** to save the file with a different name and/or path than that displayed in the **File Path/Name** field. A file browser opens for selecting the path and specifying the filename.
- ◆ Choose **File** ▸ **Save All** to save all files that have been modified in File Compare/Merge with their current names and paths.
- ◆ Choose **File** ▸ **Close Current Pane** to close the current tab pane containing a File Compare/Merge session. The **Save Results** dialog box opens and prompts you to save the files that have changed. Check the specific files you want to save and click **Yes**. The files are saved with their current name and path.
- ◆ Choose **File** ▸ **Exit** to close all the active File Compare/Merge sessions. The **Save Results** dialog box opens and prompts you to save all files that have changed. Check the specific files you want to save and click **Yes**. The files are saved with their current name and path.

Note: If you edit the temporary copy of a file revision from the StarTeam repository, unless you explicitly save that file to a different location with a new name, File Compare/Merge assigns the file a new name and saves it in a **temp** folder in your local home folder. The naming convention is the word **star**, followed by numbers (for example, **star23276.xml**). The file extension remains the same as the original file.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing a Local File with a Repository File](#)

[Comparing Historical File Contents](#)

[Comparing Two Local Files](#)

[Comparing Folders](#)

[Merging a Local File with the Tip Revision](#)

[Merging Folders](#)

[Comparing Images](#)

[Editing Files in a File Compare/Merge Session](#)

[Generating Reports from a File Compare/Merge Session](#)

[Customizing Compare and Merge Reports](#)

[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)

[File Compare/Merge Keyboard Shortcuts](#)

[File Compare/Merge Actions](#)

Setting File Compare/Merge Options

You can use the File Compare/Merge options to configure its behavior and screen display. You can only set options in the main and standalone File Compare/Merge windows. The options available depend on which version of File Compare/Merge you are using. For example, if you are using the main File Compare/Merge in the client, folder options are not available since you can not compare folders in this version. All options are available in the standalone File Compare/Merge.

To set File Compare/Merge options

- 1 Start File Compare/Merge.
- 2 On the File Compare/Merge menu, choose **Tools ▸ Options** (CTRL+O).
- 3 Select the page of options you want to change: General, File Comparison, or Folder Comparison.
There are general options, and options specific to file or folder comparisons and merges.
- 4 Make the changes you want, then click **OK**.

Note: You can reset the options to their default values by deleting the `config.properties` user settings file found in your `<username>\.fcm` folder. (For example, in Windows it would be `C:\Documents and settings\<user home>\.fcm\config.properties`).

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Merging a Local File with the Tip Revision](#)

Related Reference

[File Compare/Merge Options](#)

Working with Folders and Items

This section includes procedures related to working with folders and items.

In This Section

[Adding Folders to Views](#)

Describes how to add new folders to views

[Attaching Labels to Folders](#)

Describes how to label folders.

[Attaching Labels to Items](#)

Describes creating new revision label for an item, reviewing all labels attached to item revisions, and moving a revision label from one item revision to another.

[Changing a View's Default and Alternate Working Folders](#)

Describes how to change the Default and Alternate Working Folders for a view.

[Changing Name or Description of Folders and Items](#)

Describes the procedure for changing folder and item names and descriptions.

[Comparing Properties of Two Non-file Items](#)

Describes how to compare the properties of two non-file items.

[Configuring \(Rolling Back\) Folders and Items](#)

Describes how to configure, or roll back, a folder or item to a specific view label, promotion state, or date and time.

[Controlling EOL Characters](#)

Describes how to control whether EOL Conversion is active for a folder.

[Creating a Working Folder](#)

Describes how to change create a working folder in a StarTeam View.

[Creating Reports](#)

Describes how to create reports.

[Customizing Link Item Properties](#)

Describes how to modify item properties from a link.

[Deleting Folders and Items](#)

Describes the procedures for deleting and renaming folders and items.

[Displaying Item Details](#)

Describes how to display the item details in the lower pane.

[Displaying Location References](#)

Describes how to view folder or item location references.

[Emailing Item Properties](#)

Describes how to email item properties to another team member.

[Excluding Files and Folders from a Project](#)

Describes how to exclude files and folders from a project.

[Finding Items](#)

Describes how to find items by using field content.

[Hiding Folders and Files](#)

Describes how to make folders or files invisible in the view.

[Highlighting Items of Interest](#)

Describes how to flag items of interest.

[Linking Items Internally or Externally](#)

Describes how to link folders and items both internally in the same view, or externally between items on two different servers or configurations.

[Locking and Unlocking Items](#)

Describes how to lock and unlock items.

[Marking Items Read or Unread](#)

Describes how to flag items that you have and have not read.

[Merging Folders](#)

Describes how to merge the contents of folders.

[Moving Folders or Items](#)

Describes how to move folders and items within and between views.

[Opening a Local Folder from StarTeam](#)

Describes the procedures for opening a local folder from inside the StarTeam client.

[Renaming Files](#)

Describes the process of renaming files in StarTeam.

[Restoring Folder Selection on Tab Change](#)

Describes how to control folder selection.

[Selecting Referenced Items in Other Views](#)

Describes how to open a referenced item in a different view.

[Sharing Folders or Items](#)

Describes how to share folders and items between views on the same server..

[Sorting and Grouping Data](#)

Describes how to sort and group data.

[Viewing or Modifying Item Properties](#)

Describes how to view or modify properties for items using the standard properties dialog box.

[Viewing Previous File Revisions](#)

Describes how to view a previous file revision.

Adding Folders to Views

To add a new folder to a view

- 1 Right-click the root folder of the view and choose **New** to open the **New Folder Wizard**.
- 2 In the **New Folder Wizard**, select a parent folder for the new folder in the folder tree and click **Next**.
The new folder will be created as a child of the selected folder, and the **Folder Name** page of the wizard displays.
- 3 Type a name of up to 254 characters for the child folder.
- 4 Do one of the following:
 - ◆ Leave the **Working Folder** text box blank. The application creates the working folder using the name of the new folder and the path to its parent's working folder.
 - ◆ Type in or browse to the path for an existing working folder in the **Working Folder** text box. When you browse for a path, you create an absolute path to this folder's working folder.

Note: If you want this folder's working folder to be relative to its parent folder's working folder, type the addition to that path in this text box. Do not browse.

- 5 Type a description of up to 254 characters for the new folder in the **Folder Description** text box, and click **Next**.

The **Child Folders** page of the **New Folder Wizard** displays the new folder in the **New Folder's Child Folders** box. If the working folder has child folders, an application folder is created for each of them.

- 6 Do one of the following:
 - ◆ To exclude a child folder from your project, select the folder and click **Exclude**.
 - ◆ To exclude all child folders, click **Exclude All**.
 - ◆ To re-display folders you have excluded, click **Reset Folders**.

- 7 Click **Next**.

The **Folders** page of the **New Folder Wizard** displays the view's folder hierarchy with the new child folder.

- 8 Click **Finish**.

To add Not-in-View folders to a project

- 1 Choose **Folder Tree** ► **Show Not-in-View Folders** to make Not-In-View folders visible in the client.
With this item checked on the **Folder Tree** menu, you can see Not-in-View folders in the folder hierarchy on the left, and on the **Folder** tab in the upper pane.
- 2 Select the Not-in-View folder(s) you want add to your view.

Tip: You can select multiple sub-folders on the **Folder** tab using CTRL+CLICK, or SHIFT+CLICK+ARROW.

- 3 Right-click the selected files and choose **Add to View**.
- 4 Select the files in the folder you just added on the **File** tab and choose **Add Files**.

Note: This operation allows you to add the selected folder (and any Not-In-View parent folders) to the project folder tree. This is an alternate way to create folders instead of using the **New Folder** wizard. The folders it creates have the same folder name and working folder path.

Related Concepts

[Folders](#)

[Overview of Folders and Paths](#)

Related Procedures

[Moving Folders or Items](#)

[Attaching Labels to Folders](#)

[Attaching Labels to Items](#)

[Deleting Folders and Items](#)

[Changing a View's Default and Alternate Working Folders](#)

[Viewing or Modifying Item Properties](#)

[Viewing or Modifying Item Properties](#)

[Locking and Unlocking Items](#)

Attaching Labels to Folders

Labeling folders is slightly different from labeling items. When you attach a revision label to a folder, you can also attach it to the items that the folder contains and to everything in the subtree for which the folder is the root (its child folders and their contents).

If you detach a revision label from a folder, you can also detach the label from the items associated with the folder and, optionally, from the child folders and their items. If you detach a view label, the label is automatically detached from the items that the folder contains, from the child folders, and from their contents.

Note: To determine whether a label is a revision label or a view label, double-click the label (or select the label, and then click Properties). A revision label has a name and a description. A view label has a name, description, and a configuration time.

To create a new revision label and attach it to a folder and its contents

- 1 Select a folder in the folder hierarchy tree or on the **Folder** tab in the upper pane.
- 2 Right-click the selected folder and choose **Labels** to open the **Labels** dialog box.
- 3 Select the revision that will receive the new label.
- 4 Click **New**.
The **Attach a New Revision Label** dialog box opens.
- 5 Type a name and description for the label in the appropriate text boxes (up to 254 characters.)
- 6 Optionally, check **Frozen** (that is, cannot be changed) to ensure that only the selected revision can have this label.
- 7 Do one of the following:
 - ◆ Select **Folder Only** to attach a label to only the selected folder.
 - ◆ Select **Folder and Items Contained in Folder** to attach a label to the folder and its items.
 - ◆ Select **Everything in Subtree Rooted at Folder** to attach a label to the folder, its items, and its child folders and their items.

Note: Because attaching a label to a folder also allows it to be attached to the folder contents, children, and so on, the label is always attached to the current configuration of each folder and item. You cannot label a prior revision of a folder.

To attach an existing view or revision label to a folder and its contents

- 1 Select a folder in the folder hierarchy tree or on the **Folder** tab.
- 2 Right-click the selected folder and choose **Labels** to open the **Labels** dialog box.
- 3 Click **Attach**. The **Attach a Label** dialog box lists all the existing labels and identifies them as view or revision labels. By default, both the **View Labels** and **Revision Labels** check boxes are selected.
To display only view labels or revision labels, uncheck the appropriate check box.
- 4 Select a label.
- 5 Do one of the following:
 - ◆ Select **Folder Only** to attach a label to only the selected folder.
 - ◆ Select **Folder and Items Contained in Folder** to attach a label to the folder and its items.

- ◆ Select **Everything in Subtree Rooted at Folder** to attach a label to the folder, its items, and its child folders and their items.

Note: Attaching a label to a folder always attaches it to the current configuration of each folder and item. It is not possible to label a past revision of a folder, although you can do so for items.

6 Click **OK**.

To review the labels attached to a folder's revisions

- 1 Select a folder in the folder hierarchy tree or the **Folder** tab.
- 2 Right-click the selected folder and choose **Labels** to open the **Labels** dialog box.
The **Labels** dialog box lists all labels currently attached to this folder on a revision-by-revision basis.

To move a revision label from one folder revision to another

- 1 Select a folder in the folder hierarchy tree or on the **Folder** tab.
- 2 Right-click the selected folder and choose **Labels**.
- 3 Drag a revision label from one folder revision node to another in the **Labels** dialog box.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)
[Creating View Labels](#)
[Creating Revision Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Deleting Labels](#)
[Detaching Labels from Items](#)

Attaching Labels to Items

If you are dealing with an item or set of items that you want to group together, you can create a new revision label, attach an existing label to the item or an item revision, review all labels, or move a revision label.

To create a new revision label for selected items

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select one or more items in the upper pane.
- 3 Right-click the selected item(s) and choose **Labels** ► **New** to open the **Attach a New Revision** dialog box.
- 4 Type a name and description for the label in the appropriate text boxes. The maximum label name length is 64 characters and the description length is 254 characters.
- 5 Optionally, check **Frozen** to ensure that only the selected item revision can have this label.
- 6 Indicate what item revision is to receive this label by selecting a configuration option. The choices are:
 - ◆ **Current Configuration** to attach the label to the tip revision.
 - ◆ **Labeled Configuration** to attach the label to the revision with a specified label. The labels are in reverse chronological order based on the time at which they were created.
 - ◆ **Promotion State Configuration** to attach the label to the revision currently in a specified promotion state. (Actually, the label is attached to the revision that has the promotion state's current view label.)
 - ◆ **Configuration As Of** to attach the label to the revision that was the tip revision at a specified date and time.
- 7 Click OK.

To attach an existing view or revision label to selected items

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select one or more items in the upper pane.
- 3 Right-click the selected item(s) and choose **Labels** ► **Attach** to open the **Attach a Label** dialog box. This dialog box lists all existing labels and identifies them as view or revision labels. By default, both the **View Labels** and **Revision Labels** options are checked.
- 4 Uncheck **View Labels** or **Revision Labels** to limit the list to one specific type of label.
- 5 Select a label from the list.
- 6 Optionally, change the selection for what item revision is to receive this label by selecting a configuration option. The choices are:
 - ◆ **Current Configuration** to attach the label to the tip revision.
 - ◆ **Labeled Configuration** to attach the label to the revision with a specified label. The labels are in reverse chronological order based on the time at which they were created.
 - ◆ **Promotion State Configuration** to attach the label to the revision currently in a specified promotion state. (Actually, the label is attached to the revision that has the promotion state's current view label.)
 - ◆ **Configuration As Of** to attach the label to the revision that was the tip revision at a specified date and time.

- 7 Click **OK**.

To attach an existing view or revision label to a specific item revision

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select the item in the upper pane, then click the **Label** tab in the lower pane.
- 3 Right-click an item revision in the **Label** pane and choose **Attach** to open the **Attach a Label** dialog box. This dialog box lists all existing labels and identifies them as view or revision labels. By default, both the **View Labels** and **Revision Labels** options are checked.
- 4 Uncheck **View Labels** or **Revision Labels** to limit the list to one specific type of label.
- 5 Select a label from the list.
- 6 Optionally, change the selection for what item revision is to receive this label by selecting a configuration option. The choices are:
 - ◆ **Current Configuration** to attach the label to the tip revision.
 - ◆ **Labeled Configuration** to attach the label to the revision with a specified label. The labels are in reverse chronological order based on the time at which they were created.
 - ◆ **Promotion State Configuration** to attach the label to the revision currently in a specified promotion state. (Actually, the label is attached to the revision that has the promotion state's current view label.)
 - ◆ **Configuration As Of** to attach the label to the revision that was the tip revision at a specified date and time.

- 7 Click **OK**.

To review all labels attached to item revisions

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select the item in the upper pane, then click the **Label** tab in the lower pane. This action displays the **Label** pane which shows all revisions for the item.
- 3 Right-click a revision in the **Label** pane to display all of its labels.

To move a revision label from one item revision to another

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select the item in the upper pane, then click the **Label** tab in the lower pane. This action displays the **Label** pane which shows all revisions for the item.
- 3 Right-click a revision in the **Label** pane to display all of its labels.
- 4 Select a specific label and drag it from one revision to the another.

Note: A Label can be attached to only one revision of an item.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Folders](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Copying Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Changing a View's Default and Alternate Working Folders

Make sure that everyone is logged off from the server and that the server is locked before you change the **Default Working Folder**. It is just as critical to perform these actions as it is when you change custom fields or do anything else that affects all users.

When you change the **Default Working Folder**, not only the path to the working folder but the path to each child folder in the view may be similarly modified—not just for you, but for everyone working with that view.

Warning: Do not change the **Default Working Folder** unless you are a project administrator. These default settings affect ALL users and incorrect settings cause other users to be unable to check out StarTeam files. The default settings should only be set to the name of the folder. If you want to use a different location for your working folder than the **Default Working Folder** path, specify an **Alternate Working Folder** path.

To change the working folder

- 1 Choose **View** ► **Properties** to open the **View Properties** dialog box.
- 2 Select the **Name** tab.
- 3 Do one of the following:
 - ◆ Select **Alternate** to create a different working folder for only yourself.
 - ◆ If you are a project administrator, select **Default** to specify the default repository path for all users.
- 4 Type the name of a new working folder or browse for a path to a working folder. If you browse for the path, it becomes an absolute path. This path can be edited, however, to enable you to work on a computer that uses a different letter for its hard drive.

Note: It is important that the **Default Working Folder** point to a location that is physically discrete for each user, such as a drive on that user's workstation or a personal directory on a shared file server.

Related Concepts

[Overview of Views](#)
[Understanding Default and Alternate Working Folders](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)
[Branching Behavior of Items](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)

Changing Name or Description of Folders and Items

To change the name and/or description of a folder or item

- 1 Select the folder or item in the **Folder Tree** or upper pane.
- 2 Right-click the selected folder or item and choose **Properties**.
- 3 Change the **Name** and/or **Description** for the folder or item and click **OK**.

The folder or item name and/or description is changed in both the StarTeam repository and in your working folder.

Related Concepts

[Folders](#)

[Files](#)

[Requirements](#)

[Tasks](#)

Related Procedures

[Working with Folders and Items](#)

Comparing Properties of Two Non-file Items

To compare the properties of two non-file items, or different revisions of the same item

- 1 Select the two non-items whose properties you want to compare, such as two change requests in the **Upper** pane, or select two revisions of the same change request in the **History** pane.
- 2 Choose **Tools** ► **Compare** from the main menu.

This opens the embedded compare panes at the bottom of the windows, and displays the properties of the two selected items. You can only view the properties here. You cannot change the properties.

Related Concepts

[Overview of File Compare/Merge](#)

Configuring (Rolling Back) Folders and Items

You can configure (or roll back) an individual folder or item to a specific view label, promotion state, or date and time. Essentially, all rollbacks are made to a particular date and time. For example, if you roll back to a view label, you essentially roll back to the revision of the folder or item that existed on the date and time at which that label was attached. Unlike a view, a folder or item retains its roll-back configuration until you manually change it or until the folder or item branches. When you close the view, the folder or item does not immediately return to its current configuration.

Rolling back a folder does not re-configure any of the items or child folders associated with it. It only rolls back the folder properties to the values they had at the configuration time. Depending on the folder behavior, the folder may become read-only, in which case its properties cannot be changed.

The configuration of a folder affects the new items or child folders that can float into it. For example, in a floating branch view, you can keep items from floating into a particular folder by configuring the folder to a particular label, promotion state, or point in the past. Later, you can re-configure the folder to floating so that it can receive new items from its parent. However, the items added to the parent while the folder was not floating will never automatically go into the folder. They must be manually shared. To freeze a folder or item at a certain point in time so that it cannot be changed:

- ◆ Change its configuration to a point in the past.
- ◆ Make sure that its branching behavior is either disabled or not set to **Branch on Change**.

Warning: There is no way to locate folders that have been configured to a point in the past unless you make a note of them. Use this feature with caution.

To roll back a folder or item

- 1 Do one of the following:
 - ◆ Select a folder in the folder hierarchy tree on the left or on the Folder tab in the upper pane.
 - ◆ Select one or more items in the upper pane on any tab.
- 2 Right-click the selected items and choose **Advanced** ► **Behavior**.
This opens the **Item Behavior** dialog box.
- 3 On the **Configuration** tab, select a configuration option:
 - ◆ **Labeled Revision:** This option uses the folder or item revision with the specified view label as the tip revision. Existing view labels are listed in reverse chronological order based on the time for which they were created. This option is disabled if the view has no labels defined.
 - ◆ **Promotion State Configuration:** This option uses the folder or item revision with the view label assigned to the selected promotion state as the tip revision. This option is disabled if this view has no promotion states defined.
 - ◆ **Revision As Of:** This option uses the folder or item revision just prior to the specified point in time as the tip revision. It defaults to the current date and time, but you can select a date and time in the past, as long as it is after the time when the folder or item was created.
- 4 Click **OK**.

To return to the current folder or item configuration

- 1 Do one of the following:
 - ◆ Select a folder in the folder hierarchy tree on the left or on the Folder tab in the upper pane.
 - ◆ Select one or more items in the upper pane on any tab.
- 2 Right-click the selected items and choose **Advanced** ► **Behavior**.
This opens the **Item Behavior** dialog box.
- 3 On the **Configuration** tab, select **Floating**, then click **OK**.

To identify rolled-back items

- 1 Right-click the column headers in the upper pane and choose **Show Fields**.
- 2 Select the **Configuration Time** field in the **Available Fields** list and click **Add**.
This displays the **Configuration Time** column the upper pane.
- 3 Sort based on the **Configuration Time** column, because only items with configuration times in the past have any values in this field. If the **Configuration Time** matches the time for a view label, the item may be configured to that view label or to a promotion state based on that view label.
Other fields that may be of interest are Branch On Change and Branch State.
 - ◆ **Branch On Change** uses a **Yes** or **No** value to indicate whether a file will branch when it changes. If **Branch On Change** is **No** and **Configuration Time** has a value, the item is frozen (read-only).
 - ◆ **Branch State** indicates whether an item has branched in the child view, is still unbranched and part of the parent view, or is in the root view for the project and, therefore cannot be branched. Its values are **Branched**, **Not Branched**, and **Root**.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Displaying Additional Fields](#)
[Applying Predefined Filters](#)
[Filtering Data](#)
[Viewing or Modifying Item Properties](#)
[Viewing Previous File Revisions](#)

Controlling EOL Characters

You can specify which EOL and path case sensitivity settings are active for file check-out operations that you perform.

To control EOL settings

- 1 Choose **Tools** ► **Personal Options** and select the **File** tab.
- 2 Select or clear **Automatic EOL Conversion On Check-out**.
- 3 Click the appropriate radio button to specify whether you are working on Windows, Unix, or Mac.

Related Concepts

[Personal Options](#)

Related Procedures

[Customizing Personal Options](#)

Creating a Working Folder

In StarTeam, each of the child folders in the view has its own working folder, which is generally relative to the path of the root working folder. When you check out a file, StarTeam copies the requested file revision to the appropriate working folder. If the working folder does not currently exist on your workstation, StarTeam automatically creates it for you when you check out files.

You can also add new files to StarTeam from a working folder. If the appropriate folder does not yet exist on your workstation, you can create it automatically by using the [Create Working Folders](#) command. Once the working folder exists, you can place files in it and add them to StarTeam.

Warning: The default working folder must point to a location that is physically discrete for each user, such as a drive on that user's workstation or a personal directory on a shared file server.

To create a working folder

- 1 Select the appropriate folder in the folder hierarchy tree.

Note: A folder that exists in the StarTeam repository, but not on your local workstation is represented by a folder icon with an exclamation mark on it.

- 2 Do one of the following:

- ◆ Check out a file from the folder. StarTeam will automatically create a working folder with the same name and the same path.
- ◆ Right-click the folder and choose [Create Working Folders](#). After the working folder exists, you can copy files to it or create files in it and add them to the StarTeam repository.

Note: If the working folder path for a shared or moved folder exceeds the operating system's maximum working folder path length of 254 characters (including [\] backslashes), StarTeam will not allow you to create the working folder and displays an error message.

Related Concepts

[Overview of Views](#)
[Understanding Default and Alternate Working Folders](#)
[Proper Use of Views](#)
[Overview of Branching Options](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Views](#)
[Creating and Configuring Views](#)

Creating Reports

All reports that you create in the application show all or some portion of the data displayed in the upper pane. All reports are generated in .html format.

Note: You can also create reports using StarTeam Datamart. See the “StarTeam Datamart User Guide.pdf” for more information.

To create reports

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Choose **Reports** from the component or context menu. The **Reports** dialog displays the **Available Reports** list box.
- 4 From the **Available Reports** list box, choose the type of the report you want to generate.
- 5 Do one of the following:
 - ◆ To include only the items selected on the upper pane, select the **Current Selection** option button.
 - ◆ To include all items displayed in the upper pane, select the **Select All** option button.
- 6 Type or browse to the path and report filename in the **Output file name** text box.
Be sure to use `.htm` or `.html` as the file extension.

Note: By default, the default report filename uses the convention `<STReport><date><alphanumeric code>.html` (e.g. `STReport2006-07-24T22-03-59Z.html`).
- 7 Type a name for your report in the **Report title** text box.
- 8 Click **Generate** to view the report on screen. Your Web browser opens and displays your report. In addition, the report is saved in the location specified in step 6 above.

Related Concepts

[Reports](#)

[Data Export with Datamart](#)

Related Procedures

[Customizing Report Templates](#)

[Configuring the Report Output Path](#)

[Printing Reports](#)

Customizing Link Item Properties

You can view or modify folder and item properties directly from the **Link** pane.

To view or modify folder or item properties from a link

- 1 Select an item in the upper pane that links to a folder or another item.
- 2 Select the **Link** tab in the lower pane.
- 3 Right-click a link and choose **Item Properties** to display a **Properties** dialog box.
This dialog box displays information about the folder or item in the **Item** column in the **Link** pane.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Related Reference

[Project, Folder, and Item Properties](#)

Deleting Folders and Items

The procedures below explain how to delete folders and items in the StarTeam client. They also explain how to delete the your working folders without deleting the folders in the view or project.

To delete a StarTeam folder

- 1 Select the folder you want to delete in the project view.
You can do this in the folder hierarchy tree on the left, or on the **Folder** tab in the upper pane.
- 2 Right-click the selected folder and choose **Delete**.
A dialog box opens prompting you to confirm the deletion.
- 3 Optionally, check **Delete Working Folders** to include your local folder and its contents.
- 4 Click **Yes** to delete the folder.

StarTeam deletes the selected folder and all its files and sub-folders.

To delete a local folder only

- 1 Select the folder you want to delete in the project view.
You can do this in the folder hierarchy tree on the left, or on the **Folder** tab in the upper pane.
- 2 Right-click the selected folder and choose **Delete Local Folders**.
A dialog box opens prompting you whether you really want to delete the selected folder and all its files and sub-folders.
- 3 Click **Yes**.

To delete an item

- 1 Click the tab in the upper pane containing the item or items you want to delete: **File**, **Change Request**, **Requirement**, **Task**, or **Topic**.
- 2 Select one or more items to delete.
- 3 Right-click the selected item(s) and choose **Delete**.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Displaying Item Details

You can view an item's details quickly using the **Detail** tab in the lower pane. The Detail pane displays the item properties and their values in a two-column list.

The properties that display in the **Detail** pane for file, change request, child folder, and audit items are determined by which fields you choose to display in the upper pane, and/or which filters you use for the component tab.

To display item details

- 1 Click a component tab in the upper pane.
- 2 Select an item in the upper pane.
- 3 Click the **Detail** tab in the lower pane.

Related Procedures

[Displaying Additional Fields](#)

[Applying Predefined Filters](#)

Displaying Location References

Because of manual sharing and because views are children of other views, a folder or item can be associated with more than one project, view, or parent folder (within the same server configuration). Each instance of the folder or item has a reference to its tip revision.

To view folder references, you open a separate dialog; to view item references, you use the project view window and the **Reference** tab on the lower pane.

To view references, do one of the following

- 1 View folder references
- 2 View references for past revisions of a folder
- 3 View item references
- 4 View references for past revisions of an item

To view folder references

- 1 Select the folder in the folder hierarchy tree.
- 2 Right-click the selected folder and choose **Advanced** ► **References**.

The **Folder References** dialog box opens and displays a tree that indicates which project views reference this folder, and their relationship to each other.

View references for past revisions of a folder

- 1 Right-click the folder in the folder hierarchy tree and choose **Properties**.
- 2 Click the **History** tab.

Note: There is no **History** tab if you do not have the access rights that allow you to see the folder history.

- 3 Select the revision in the **History** list.
- 4 Right-click the selected revision and choose **References**.

An appropriate **References** dialog box appears.

To view item references

- 1 Select an item in the upper pane.
- 2 Click the **Reference** tab beneath the lower pane.

Data similar to that in the **Folder References** display in the lower pane. The **Reference** pane has no context menu.

View references for past revisions of an item

- 1 Select an item in the upper pane.
- 2 Click the **History** tab in the lower pane.

- 3 Select the revision in the **History** list.
- 4 Right-click the selected revision and choose **References** from the **History** context menu.
An appropriate **References** dialog box appears.

Related Concepts

[References to Folders and Items](#)

Related Procedures

[Managing Projects](#)

[Working with Folders and Items](#)

Emailing Item Properties

You can send a text representation of selected items (except files) as an email message, along with additional text. The information sent for each item includes the fields displayed in the upper pane. For items such as change requests, the item's properties, which are the same as its contents, are sent in the email. For files, only the properties can be sent. However, a shortcut to the item can be included.

Items are considered to have been sent by the application, not by you. Therefore, you may want to copy yourself on the email. Otherwise, you will not receive the message.

Note: If you set up a filter in the client and email an item, only the fields displayed by the filter are sent to the recipient.

To send item properties by email

- 1 Select a folder in the folder hierarchy, click a component tab, and select an item in the upper pane.
- 2 Do one of the following:
 - ◆ Click the **Send** button on the toolbar. (If you have selected a file component, this button does not appear.)
 - ◆ Right-click the selected item and choose **Send To**.

This opens the **Send To** dialog box.

- 3 Click **To** or **CC** to open a dialog box for selecting the primary or secondary email recipients.
Select the email recipients by moving the team member names from the **Available Users** to **Selected Users** list and click **OK**.
- 4 Type a **Subject**.
- 5 Optionally, check **Send A Copy To Myself** if you want to receive a copy of the email.
- 6 Optionally, check **Attach Item Shortcut** to include a shortcut to this specific item in the email.
- 7 Type any additional information in the **Add Text To The Mail Message** text box.
- 8 Click **Send Now** to send the message.

Unlike automatic email notification, this message will not display the word “notification” in the subject line.

Note: Do not confuse email messages sent by individuals with email notification messages automatically sent by the server. If your administrator has enabled email notification, you will automatically receive email messages notifying you about items for which you are responsible and topics for which you are listed as a recipient.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Properties](#)
[Comparing Historical File Contents](#)
[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Excluding Files and Folders from a Project

Some types of files will never be added to a project, although they may reside in a working folder. For example, suppose you are creating files with an application that makes an automatic backup (.bak) copy of each file every time you save the file. Although your working folder might contain several .bak files, you would have no reason to check them into (or out of) the application. Therefore, you should exclude them from the project view.

Exclude lists can also be inherited from parent folders.

To exclude files from a project

- 1 Select a folder from the application folder hierarchy tree, or from the **Folder** tab in the upper pane.

Note: You can also exclude files in Not-in-View folders, but you must have the root folder selected to do so.

- 2 Right-click the selected folder and choose **Properties** to open the **Folder Properties** dialog box.
- 3 Select the **Exclude** tab.

Note: The **Exclude** tab does not affect files that are already part of the project.

- 4 Choose one of the following:

- ◆ **Inherit and Use Local Exclude List:** Excludes files that match the exclude list specifications set for this folder as well as those of its parent folder. If the **Local Exclude List** text box does not yet include any file specifications, add them.
- ◆ **Use Local Exclude List:** Excludes files that match the exclude list specifications set for this folder. If the **Local Exclude List** text box does not yet include any file specifications, add them.
- ◆ **No Exclude List:** includes all files.

- 5 Type one or more file specifications to use for matching files.

Use the standard expressions (with * and ? wild cards) separated by commas, spaces, or semicolons. To include a comma, space or semicolon as part of the specification, enclose the specification in double quotes.

A trailing / character represents Not-in-View folders to be excluded. For example, bin/ would cause all Not-in-View folders named bin to be excluded from the folder tree.

Note: The \ character does not work; it is treated as an escape character.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Finding Items

You can search all items displayed in the upper pane for the data contained in any displayed field. For example, you can locate a change request by its number or search for a file with a particular name, status, time stamp, or size.

If you want to search a field that is not displayed, double-click a column header in the upper pane, select **Show Fields** from the context menu, select the desired field, and click **Add**.

To find items

- 1 Select a folder in the folder hierarchy tree.
- 2 Click a component tab in the upper pane.
- 3 Right-click in the upper pane and choose **Find**.
The **Find** dialog box opens.
- 4 Type part or all of the data in the **Search For** text box.
Do **not** use wild cards.
- 5 Select **Forward** to search the upper pane from the top to the bottom, or select **Backward** to search the upper pane from the bottom to the top.
- 6 Select **Starting at: Currently Selected Item** to begin searching from the item that is presently selected, or select **Starting at: First Item** to search from the first item in the upper pane.
- 7 Select either **All Displayed Fields** or **This Field**. If you select **This field**, select the field for which you want to search from the drop-down list box.
- 8 Check **Match Case** if a case-sensitive search is appropriate.
- 9 Click **Find** to search.

Tip: Use **CTRL+F** to start a search. Use **F3** to find the next item that matches the search text and **SHIFT+F3** to find the previous item that matches the search text

Related Procedures

[Working with Folders and Items](#)

Hiding Folders and Files

Using the **Folder Properties** dialog box, you can set the **Visible** property to exclude folders and their files from visibility.

To hide a folder

- 1 Do one of the following to open the **Folder Properties** dialog box:
 - ◆ Double-click the folder on the **Folder** tab.
 - ◆ Right-click the folder in the **Folder Tree** and choose **Properties**.
- 2 Select the **Name** tab In the **Folder Properties** dialog box and uncheck the **Visible** option to hide the folder and the files it contains.
- 3 Click **OK** to close the dialog box and save the folder properties.

Note: To make the folder visible again, check the **Visible** option in the **Folder Properties** dialog box.

Related Concepts

[Overview of Folders and Paths](#)

[Files](#)

[Folders](#)

Related Procedures

[Excluding Files and Folders from a Project](#)

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Highlighting Items of Interest

To highlight specific items (except audit entries) on the upper pane, you can add a flag to them. For example, you might wish to flag items related to a particular customer request.

Flags are set, viewed, and removed by the user who created them. If an item has been flagged, the **Flag** field displays **Yes**. If an item is not flagged, the **Flag** field displays **No**. To display flagged items, the upper pane must be in list format.

To flag an item

- 1 Select a folder in the folder hierarchy tree.
- 2 Select any component tab except the **Audit** tab.
- 3 Select the item you want to flag.
- 4 Choose **Flag** from the corresponding item menu or the context menu.

If the items in the upper pane appear in tree format, switch to the list format to display the flagged items.

Note: If the **Flag** field does not appear, right-click a column header, and choose **Show Fields** from the context menu. Select the **Flag** field from the **Available Fields** list, click **Add**, then click **OK**.

To remove a flag from an item

- 1 Select a folder in the folder hierarchy tree.
- 2 Select any component tab except the **Audit** tab.
- 3 Select the item you want to flag.

If the items in the upper pane appear in tree format, switch to the list format to display the flagged items.

- 4 Choose **Remove Flag** from the corresponding item menu or the context menu.

Tip: Use the following shortcut keys: **CTRL+F2** to flag an item; **CTRL+SHIFT+F2** to remove a flag from an item.

Linking Items Internally or Externally

This procedure describes how to link two items, either internally in the same server configuration, or linking between two items located on different server configurations, called *external linking*.

In StarTeam, an *item* is a file, change request, requirement, task, or topic. A *link* is a connection between two folders, two items, or a folder and an item. Creating links can be quite useful. For example, linking a file to a change request allows you to mark it as fixed when you check in the edited file. By linking files to the requirements document that the files fulfill, you can easily refer to or update the document.

You can create several links at the same time if you want to link several items of the same type to one particular item. For example, you might wish to link several change requests to a single file. To accomplish this, you can create links using the **Folder Tree** menu, component menu, context menu, or **Link** button on the toolbar.

Note: When you create external links between items on different server configurations, both server configurations need to be opened in the Cross-Platform Client to be able to create or view the external links.

To link a folder or item(s) to one or more items

- 1 Begin the link process by doing one of the following:
 - ◆ Open the server configuration and view which contains the items you want to link. If you want to link two items on different server configurations, open both server configurations and views in the Cross-Platform Client.
 - ◆ Select a folder in the folder hierarchy or in the upper pane on the **Folder** tab.
 - ◆ Click a component tab in the upper pane, such as **File**, **Change Request**, **Requirement**, **Topic**, or **Task**, and select one or more items.

Note: You cannot link two folders to each other with the method described in this procedure.

- 2 Right-click the selected item(s) on the component tab and choose **Links** ► **Create Link**.
The **Links** menu is also available on the component menu that corresponds with the selected component tab.
This action changes the mouse pointer and displays it as a knotted rope.

Note: If you initially select an item from the upper pane, you can also use the **Link** button on the toolbar; however, this button is disabled if you start the link with a folder.

- 3 Select the folder or item(s) for the end of the link from the same configuration or from the external server configuration. This can be:
 - ◆ A StarTeam folder (if you have not already selected a folder).
 - ◆ One or more other files.
 - ◆ One or more change requests.
 - ◆ One or more requirements.
 - ◆ One or more topics and/or responses.
 - ◆ One or more tasks and/or subtasks.

To locate all items, you may need to switch to a different component tab or use the **All Descendants** button on the toolbar.

- 4 Choose **Links** ▸ **Complete Links** on the **Folder** menu, the component menu, or the context menu, or click the **Link** button again on the toolbar.

This button is disabled if you are linking an item to a folder.

- 5 Verify that the links exist by doing one of the following:

- ◆ Select a linked item, then click the **Link** tab on the lower pane to view the links for the item.
- ◆ Right-click a linked folder, then choose **Properties** to display the **Folder Properties** dialog box. Click the **Link** tab to view the link. (The **Link** tab will not appear in this dialog box if you do not have access rights to view links.)

You can also view a link by selecting either of its ends. The end you select, whether a folder or an item, is called the source. The other end of the link is called the target and is listed in the **Item Type** column on the **Link** pane.

Tip: If you change your mind about creating a link after you have started to create it, but before you have finished completing it, you can select **Links** ▸ **Cancel Link** on the **Folder Tree** menu, the component menu, or the context menu. If you are using the **Link** button on the toolbar, press **ESC**.

Related Concepts

[Links: Internal and External](#)

Related Procedures

[Linking Specific Revisions](#)

[Checking Linked Files In and Out](#)

[Selecting Linked Files](#)

[Deleting Links](#)

[Linking Files to Process Items](#)

[Creating External Links](#)

Locking and Unlocking Items

Before changing the contents of a file or editing item properties, you should exclusively lock the file or item. This action informs other team members that you intend to make changes. Files, change requests, requirements, tasks, and topics can all be locked.

Exclusively locking an item prevents others from creating new revisions of it before the lock has been released. You can lock and unlock any type of item as a separate operation. In addition, you can lock and unlock files as part of the check-in and check-out processes.

If an item is exclusively locked by someone else, you can review its properties but cannot change them. Normally the words **Read Only** and the name of the user who has locked the item will appear on the title bar.

To lock an item using the toolbar

- 1 Select one or more items in the upper pane.
- 2 Click the **Lock** button on the toolbar.

The selected items become exclusively locked, and you are listed as the user who has locked them.

To lock an item using a menu

- 1 Select a folder in the folder hierarchy tree on the left.
- 2 Click any component tab in the upper pane except **Audit**. (Audit items cannot be locked.)
- 3 Right-click an item in the upper pane and choose **Lock/Unlock**.
This opens the **Set My Lock Status** dialog box.
- 4 Select a lock status option:
 - ◆ **Unlocked**: Removes your exclusive or non-exclusive lock on the selected items
 - ◆ **Exclusive**: Prevents others from creating new revision of this item (until you release the lock or another person breaks your lock).
 - ◆ **Non-exclusive**: Indicates that you are working on the item and may possibly make changes (not recommended for items other than files).
- 5 Optionally, check **Break Existing Lock** to break another team member's lock on the item.
If e-mail is enabled, StarTeam will send an e-mail message to the team member whose lock has been broken to inform him or her of this fact.

Note: You must be granted the appropriate privileges to be able to break another person's locks.

To remove your lock from an item

- 1 Select the locked item.
- 2 Click the **Unlock** button on the toolbar

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Marking Items Read or Unread

Change requests, requirements, tasks, and topics that you have not read display in boldface type if you are the user responsible for the item, or if you are the recipient of the topic. After you have reviewed the item properties, the boldface type is replaced by regular type.

To reread all the items on a certain subject, it is sometimes convenient to mark all of the items unread to ensure that you do not miss any. You can also mark an entire requirement, task, or topic tree as unread.

To mark specific items as read or unread

- 1 Select a folder in the folder hierarchy tree.
- 2 Click the **Change Request**, **Requirement**, **Task**, or **Topic** tab.
- 3 Select one or more items in the upper pane that you want made bold (unread) or not bold (read).
- 4 Right-click the selected items and choose **Mark As Unread** or **Mark As Read**.

The items are marked as specified, and the type style changes accordingly.

Related Procedures

[Working with Folders and Items](#)

[Working with Change Requests](#)

[Using Topics](#)

[Using Tasks](#)

[Using Requirements](#)

[Working with Properties](#)

Merging Folders

Merging folders is quite similar to merging files, except you are merging folder contents instead of file contents. However, if two files in a folder have differences, you can open a file comparison on those files from the folder comparison pane.

You can only merge folders using the standalone File Compare/Merge.

To merge two folders

- 1 From the Windows desktop, choose **Start ▶ Program Files**, navigate to your StarTeam program installation group, and choose **File Compare Merge**.
- 2 Choose **File ▶ New Folder Comparison** (CTRL+SHIFT+N), or **File ▶ New Folder Merge** (CTRL+SHIFT+M) to open a new folder comparison tab.

Note: Using the **New Folder Merge** command opens the folder comparison results with **Auto Commit Off**. To open a folder comparison with **Auto Commit On**, choose **File ▶ New Folder Comparison**. Once File Compare/Merge is opened, you can click **Auto Commit Mode** on the toolbar to toggle between **Auto Commit On** or **Auto Commit Off**.

- 3 Click the **Open New Comparison** buttons above each editor pane, and browse to the folders you want to compare.

Alternatively, you can click the **DOWN ARROW** to the right of the **Open New Comparison** button and select a folder from the history list, or you can type the path for each folder in the directory to the left.

By default, File Compare/Merge immediately compares the selected folders, and the results display in the two panes of the File Compare/Merge window. You can change this and other default settings in the File Compare/Merge Options dialog box on the **General** and **Folder Comparison** pages.

- 4 Optionally, right-click and choose **Switch Comparison Modes** to choose a different comparison mode display. If you choose one of the **3-Way** comparison modes, you will have a third pane for viewing and editing the merged base folder.

- 5 Perform any required changes to the folders directly in the edit panes. You can move, delete, and copy folders from one pane to the other, and are prompted to commit your changes.

The toolbar and context menus are different for folder compare/merge than for file compare/merge. For folder comparisons, you have items for moving or copying files and folders left or right, deleting files and folders from the right or left panes, and several options for expanding/collapsing tree nodes.

- 6 Optionally, double-click a file with differences to open a new File Comparison pane and resolve the differences.
- 7 Optionally, right-click in one of the panes and choose **Generate Report**.

To print the report, choose **File ▶ Print**.

Note: If you want a report of your changes, you must generate it before closing the File Compare/Merge window.

- 8 Click **Save Changes To Disk** (check mark) on the toolbar to save your changes when you are finished.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Folders](#)
[Comparing Images](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Merging a Local File with the Tip Revision](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)

Related Reference

[File Compare/Merge Keyboard Shortcuts](#)
[File Compare/Merge Actions](#)

Moving Folders or Items

Folders and items, such as files (including Not in View files) and change requests, can be moved from one project view to another as long as the two views are on the same server configuration. Moving a folder also moves its contents, its child folders, and their contents. When an item is moved to another project view, it belongs to the new view, although its behavior, configuration, and other properties do not change. It loses any labels it had in the previous view, however, because labels cannot move from view to view. Also, if you roll back the view to an earlier point in time, you will no longer see the folders and/or items that have been moved.

Moving a folder or item within a view causes that folder or item to be copied in that view's child or parent views, if branching has not occurred. In this application, a move is a copy operation followed by a delete operation, and delete operations are not propagated from view to view for folders and items that have not branched. Therefore, the view in which the move was made has one copy of the folder or item in the new location, while the related views have two copies of the folder or item, one in the original location and one in the new location — the equivalent of a share.

Note: When you move a folder or an item, the access rights set at the folder or item level accompany it. Also, in some cases, moving a folder or item to another view enables its disabled **Branch on Change** check box.

To move a folder or item within the same view

- 1 Choose **Project** ► **Open** to open the **Project View** window.
- 2 Drag the folder or item that is to be moved from one location in the view to another.
A message box appears, asking you to confirm this change.
- 3 Click **Yes**.

To move a folder or item between two different views

- 1 Choose **Project** ► **Open** to open the **Project View** window.
- 2 Choose **Project** ► **Open** to open a second project in another window.
Make sure both project views are visible. Use one of the **Window** menu commands (**Cascade**, **Tile Vertically** or **Tile Horizontally**) to this, and resize the windows if required.
- 3 Drag the folder or item that is to be moved from one view to the other view.
A message box appears, asking you to confirm this change.
- 4 Click **Yes**.

Warning: You cannot move tasks and subtasks that have been exported from MS Project to StarTeam.

Related Concepts

[Folders](#)

Related Procedures

[Adding Folders to Views](#)

[Attaching Labels to Folders](#)

[Sharing Folders or Items](#)

[Attaching Labels to Items](#)

[Deleting Folders and Items](#)

[Changing a View's Default and Alternate Working Folders](#)

[Configuring \(Rolling Back\) Folders and Items](#)

[Viewing or Modifying Item Properties](#)

[Locking and Unlocking Items](#)

Opening a Local Folder from StarTeam

The following procedures describe how to quickly open a local folder in a file browser from within the StarTeam client so you can perform basic file and folder management tasks.

To open a local folder from a folder selection in StarTeam

- 1 In StarTeam, select from one to five folders you want to open.
You can select one folder in the folder hierarchy tree on the left or up to five folders on the **Folder** tab in the upper pane.
- 2 Right-click the selected folder and choose **Open Local Folder**.

This opens a **Windows Explorer** for each location on disk that corresponds to a selected folder. This applies to all folders except those whose status is **Missing** since their local folders do not exist.

To open a local folder from a file selected in StarTeam

- 1 On the **File** tab in the upper pane, select one to five files.
- 2 Choose **File** ► **Open Containing Folder**.
This opens a **Windows Explorer** to each folder in which a selected file exists. This applies to all folders except those whose status is **Missing** since their local folders do not exist.

Related Concepts

[Folders](#)

[Files](#)

Related Procedures

[Working with Folders and Items](#)

Renaming Files

If you want to rename a file in your project, you should rename it within the StarTeam application. This retains the properties associated with that file, such as history and links. However, when you roll a view or item back to a configuration prior to the name change, the old name is not restored. This can cause problems. For example a setup script from an earlier build or release might fail because it uses the file's former name.

If you rename a file outside the application (for example, by using Windows Explorer), the application considers the file to be a new file. When you add the file with the new name, it will have no connection to the history, links, or other properties of the original file.

To rename a file

- 1 Select the file you want to rename in the upper pane.
- 2 Do one of the following:
 - ◆ Choose **File** ► **Properties**.
 - ◆ Right-click the selected file and choose **Properties**.
 - ◆ Click the **Properties** toolbar button.
- 3 Click the **General** tab in the **File Properties** dialog box.
- 4 Change the **Name** and click **OK**.

StarTeam renames the file in both the StarTeam repository and in your working folder.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Restoring Folder Selection on Tab Change

When you click a new component tab in the upper right pane, StarTeam gives you the option of retaining the folder currently selected from the folder hierarchy or replacing it with the root folder.

To control the folder that displays after a tab change

- 1 Choose **Tools** ► **Personal Options** to open the **Personal Options** dialog box.
- 2 Select the **Workspace** tab.
- 3 Do one of the following:
 - ◆ Check **Restore Folder Selection on Tab Change** to retain the currently selected folder when the component tab changes.
 - ◆ Uncheck **Restore Folder Selection on Tab Change** to select the root folder every time the component tab changes
- 4 Click **OK**.

Related Concepts

[Folders](#)

[Personal Options](#)

Related Procedures

[Adding Folders to Views](#)

[Viewing or Modifying Item Properties](#)

[Attaching Labels to Folders](#)

[Moving Folders or Items](#)

[Attaching Labels to Items](#)

[Deleting Folders and Items](#)

[Changing a View's Default and Alternate Working Folders](#)

[Configuring \(Rolling Back\) Folders and Items](#)

[Locking and Unlocking Items](#)

Selecting Referenced Items in Other Views

The Reference tab located in the lower pane, provides a context menu containing commands for viewing the properties of an item and for opening a referenced item in a different view.

To open referenced items in a different view

- 1 Select the item from the upper pane.
- 2 Select the Reference tab in the lower pane.
- 3 Right-click the reference for the item, and choose **Select Referenced Item** from the context menu.

The view containing the item opens (in addition to any views that are already open) with the item selected. If the referenced view is already open, the Cross-Platform client opens an additional copy of the view so that it does not disturb anything that may have already been set up within that view.

Related Concepts

[References to Folders and Items](#)

Sharing Folders or Items

You can share folders and items, such as files or change requests, between views if the views belong to projects that are located on the same server configuration. You can also share a folder or item at two locations in the same view.

Branching a view negates all shares, not just the ones between parent and child views.

Keep in mind the following points about sharing folders or items:

- ◆ When you branch a view, any manual shares between items in the same view are not retained in the view's child view.
- ◆ When a folder is shared, users of both views can access its contents, including its child folders and their contents.
- ◆ As a rule, the behavior of the shared item is governed by the **Set items Shared Into View to Branch on Change** property of the new view, not by that of its parent view. If the new view is a reference view (that is, a view that does not permit branching), it does not have that property, so the shared item's Branch on change setting is controlled by the setting of its parent view.
- ◆ The "branch on change" behavior of a shared item is specific to the folder it is in. If the folder is in a project that resides on a 2005 Release 2 server, the "branch on change" checkbox is unchecked by default for the shared file. However, if the project resides on a 2006 or 2008 server, then the "branch on change" checkbox is checked by default for the shared file.
- ◆ Tasks and topics do not have branching behavior so this view property does not affect them.
- ◆ If a folder or item is shared, its configuration (floating, based on a label, a promotion state, or a point in time) is initially identical in both views. However, the configuration can be modified in either view, which means that shared items can differ. To freeze shared folders or items, you must configure each of them to a specific date and time. Once items have been shared into a view, you can change their branching behavior on an item-by-item basis.
- ◆ Shared folders or items have the access rights originally set for them at the folder or item level, until they branch in their new location. Branching creates a new object that initially has no access rights at the folder or item levels.
- ◆ The shared folder or item loses any labels it had in the previous view. Labels cannot be moved from view to view. However, the shared folder or item will have all the labels you attach to them in their new location regardless of whether they branch or not.

To share a folder or item in two locations in the same view

- 1 Choose **Project** ► **Open** to open the **Project View** window.
- 2 Press **CTRL** and drag the folder or item that is to be shared from one location in the view to another.
A message box appears, asking you to confirm this change.
- 3 Click **Yes**.

To share a folder or item between two different views

- 1 Choose **Project** ► **Open** to open the **Project View** window.
- 2 Choose **Project** ► **Open** to open a second project in another window.
Make sure both project views are visible. Use one of the **Window** menu commands (**Cascade**, **Tile Vertically** or **Tile Horizontally**) to this, and resize the windows if required.

- 3 Press **CTRL** and drag the folder or item that is to be shared from one view to the other view.
A message box appears, asking you to confirm this change.
- 4 Click **Yes**.

Warning: When you share tasks that have been exported from MS Project, you must share an entire task tree, starting with the root task.

Related Concepts

[Folders](#)

Related Procedures

[Adding Folders to Views](#)
[Attaching Labels to Folders](#)
[Moving Folders or Items](#)
[Attaching Labels to Items](#)
[Deleting Folders and Items](#)
[Changing a View's Default and Alternate Working Folders](#)
[Configuring \(Rolling Back\) Folders and Items](#)
[Viewing or Modifying Item Properties](#)
[Locking and Unlocking Items](#)

Sorting and Grouping Data

You can choose to do a primary sort in the upper pane (based on one column), or a more complicated sorts up to a fourth order.

To do a primary sort on one column

- 1 Open the view on the data you wish to sort or group.
- 2 Click a column header to sort the data in the upper pane based on the value in that column.
The sort is in ascending order by number, letter, internal order, or internal key, depending on the data.
- 3 Click the column header again to reverse the sort order.

A triangle appears on column header of the sorted column. The triangle points upward for ascending sorts and downward for descending sorts.

Note: You can also sort the data in the lower pane when the **Link** tab is selected.

To perform up to a fourth-order sort

- 1 Do one of the following:
 - ◆ Right-click a column header on upper pane and choose **Sort and Group** from the context menu.
 - ◆ Right-click in the upper pane and choose **Filters** ► **Sort and Group** from the context menu.

The **Sort and Group** dialog box displays four group boxes, each indented slightly more to the right than the one above it. The first group box designates a primary sort order, the second designates a secondary sort, and so on.

- 2 Optionally, check the **Show Advanced Fields** check box at the bottom of the dialog box to list all the fields in **First By** and **Then By** drop-down list boxes. Some fields are rarely used and considered advanced.
- 3 Select a field from the **First By** drop-down list box.
If you are grouping the items, the field does not need to be displayed in the upper pane. If you are not grouping the items, you can sort them based on a field that is not displayed, but you will not be able to tell where one group leaves off and the next begins.
- 4 Select the **Ascending** or **Descending** option button. The default setting is ascending order.
- 5 Select **Group By** to group the items which have the same values in this field.
If you do not select any additional sort options, text fields are sorted in ASCII order. Enumerated and user ID fields are sorted by their internal order or internal keys. That is, enumerated fields are sorted in the order given to them by the person who created the field; user ID fields are sorted in the order in which they were created. The application disables the Sort Options button for numeric and date/time fields.
- 6 Optionally, click **Sort Options** for additional sorting selections. The **Sort Options** dialog box appears.
 - ◆ Select **As Text** to sort enumerated and user ID fields by the names of their possible values. For text fields, **As Text** is your only choice.
 - ◆ Uncheck the **Case-sensitive** check box to sort alphabetically or check it to sort in ASCII order (where uppercase letters precede lowercase letters).
- 7 Add secondary and lower order sorts by using the **Then By Group** boxes as needed.

Related Concepts

[Cross-Platform Client Overview](#)

[Displaying Additional Fields](#)

Related Procedures

[Filtering Data](#)

[Creating Filters](#)

[Creating Queries](#)

[Creating Reports](#)

Viewing or Modifying Item Properties

This section explains how to use the standard properties dialog to edit item properties. Depending on how your team has set up the application, you may see a totally different dialog box called an alternate property editor (APE).

Every time the properties of an item are modified, a new revision of that item is created. If you modify a property, you should also create a revision comment explaining the modification using the **Revision Comment** tab.

To view or modify item properties

- 1 Click a folder in the folder hierarchy tree, or click one of the component tabs in the upper pane and select an item.
- 2 Do one of the following:
 - ◆ Right-click the folder or item and choose **Properties**.
 - ◆ Right-click an item on the **Reference** tab in the lower pane and choose **Item Properties**.
 - ◆ Click the **Properties** toolbar button.
- 3 Modify any of the property fields in the corresponding **Properties** dialog box that opens, then click **OK**.

Tip: You can force Java to not cache Windows file properties by placing the following option in the CPC stjava file:

```
-Dsun.io.useCanonCaches=false.
```

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)
[Viewing or Modifying Item Properties](#)

Related Reference

[Change Request Properties](#)
[File Properties](#)
[Folder Properties](#)
[Requirement Properties](#)
[Task Properties](#)
[Topic Properties](#)

Viewing Previous File Revisions

You can review the contents of a prior file revision in either the default editor or in the application for which the file type is registered.

To review a revision's contents

- 1 On the **File** tab, select the file for which you want to review a revision.
- 2 On the **History** or **Label** tab in the lower pane, select the specific revision you want to review.
- 3 Right-click the selected item to open the context menu and choose one of the following:
 - ◆ **View Revision Content** to copy the revision to a temporary file and display it in the default editor (Notepad or the alternate editor specified in the **Personal Options** dialog box).
 - ◆ **Open Revision Content** to copy the revision to a temporary file and display it in the associated application.

Note: The client creates the temporary files in the local temp directory on the system. For example, if working on a Windows system, the temporary files are created in the `C:\Documents and Settings\<user>\Local Settings\Temp` directory. When you exit the client, the files are deleted from the system.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Managing Files

This section contains tasks related to managing files.

In This Section

[Checking In Files](#)

Describes the file check-in process in StarTeam.

[Checking Out Files](#)

Describes the file check-out process in StarTeam.

[Adding Files to Projects](#)

Describes how to add from your workstation to a project folder.

[Checking Out Files in Batches](#)

Describes how to use the Bulk Check-out utility (BCO) to check out files in batches.

[Checking Out Historical Versions of Files](#)

Describes how to check out a previous revision of a file.

[Controlling How File Status Information is Stored](#)

Describes how to specify where file status information is stored.

[Deleting Folders and Items](#)

Describes the procedures for deleting and renaming folders and items.

[Displaying Item Details](#)

Describes how to display the item details in the lower pane.

[Editing Check-in Comments](#)

Describes how to edit check-in comments after the check-in has been done.

[Enabling Concurrent File Editing](#)

Describes how to control file locking from application integrations.

[Excluding Files and Folders from a Project](#)

Describes how to exclude files and folders from a project.

[Finding Files Linked to Active Process Items](#)

Describes how to find all the files linked to an active process item.

[Hiding Folders and Files](#)

Describes how to make folders or files invisible in the view.

[Locking and Unlocking Items](#)

Describes how to lock and unlock items.

[Marking Unlocked Files Read-only](#)

Describes how to mark unlocked files read-only.

[Monitoring Check-out Statistics using Cache Agent](#)

Describes how to monitor check-out statistics when using Cache Agent.

[Opening a Local Folder from StarTeam](#)

Describes the procedures for opening a local folder from inside the StarTeam client.

[Opening and Editing Files](#)

Describes how to open and edit files in StarTeam.

[Renaming Files](#)

Describes the process of renaming files in StarTeam.

[Selecting Linked Files](#)

Describes how to select linked files from a link item.

[Setting File Storage Options](#)

Describes how to control where and how file status information is stored

[Setting the File Executable Bit for UNIX](#)

Describes the task content in a sentence; used for part descriptions only.

[Using the File Annotation Tool](#)

Describes how to use the File Annotation tool, which allows you to view historical information about changes made to any text file in StarTeam.

[Viewing or Modifying Item Properties](#)

Describes how to view or modify properties for items using the standard properties dialog box.

[Viewing Previous File Revisions](#)

Describes how to view a previous file revision.

Checking In Files

To check in files

- 1 On the **File** tab in the upper pane, select one or more modified files to check in.
- 2 Choose **File** ► **Check In** to open the **Check In** dialog box.
- 3 Type a generic reason for the check-in comment, or check **Prompt for a comment (check-in reason) for each file** to open a separate **File Description** dialog box for each file.
- 4 Optionally, click **Compare** to compare the file you are checking in with the tip revision of the file in the repository. If differences exist, the **File Compare Merge** window opens showing the file differences.
- 5 Optionally, select a **Lock status** option.
 - ◆ **Unlocked:** Releases your lock on the files after check-in.
 - ◆ **Exclusive:** Indicates that you intend to make further changes to the files.
 - ◆ **Non-exclusive:** Indicates that you are working on the files and may possibly make changes.
 - ◆ **Keep current:** Retains the current lock status.
- 6 Continue completing the **Check In** dialog box options as follows.
 - ◆ Optionally, check **Force check-in** to check in files regardless of their status.
 - ◆ Optionally, check **Delete working files** to remove the selected files from your working folder after they are checked in.
 - ◆ (Required if process rules are enforced) Check **Link and pin process item** to link the new files to process items. To use a process item besides the active process item, click **Select** and use the **Select Process Item** dialog box to change the process item.
 - ◆ If work on the active process item is now complete, check **Mark selected process item as fixed/finished/complete**.
 - ◆ To make changes to the selected process item's properties during the check-in process, check **Show property editor for selected process item**.
 - ◆ Optionally, select a **Revision label** from the drop-down list, or create a new revision label by typing its name. Existing labels are listed in reverse chronological order, based on the time at which they were created, unless the **Sort View Labels By Name** option has been selected in the **Personal Options** dialog box.
- 7 Optionally, click **Advanced** to open the **Advanced Options** dialog box.
 - ◆ Check **Set EOL check-out format (for text files only)** to control the EOL character stored with the files. The default setting is based on the **EOL** setting in the **File Properties** dialog box.
 - ◆ Select an appropriate check-out **File Encoding** from the drop-down list.
 - ◆ Select the file type of file you have selected: ASCII, Binary, or Unicode.
 - ◆ Click **Show Change Requests** to review the change requests linked to the files you are checking in.
 - ◆ Click **OK** to return to the **Check In** dialog box.
- 8 Click **OK** to check in the files.

Note: You can also use the **Check In** or **Check In And Unlock** buttons on the toolbar to check in files without using the **Check In** dialog box. If process rules are required, the **Check In** dialog box will open automatically.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Checking Out Files

This procedure describes how to check out the tip revision of a file and how to check out a previous revision of a file.

To check out files

- 1 Select one or more files to check in on the **File** tab in the upper pane.

Note: You can check out files with a status of **Current**, **Out Of Date**, or **Missing**. You can also check out files with a status of **Modified**, however, you will be warned that continuing the check-out will overwrite files in your local working folder with the tip revision in StarTeam.

- 2 Choose **File** ► **Check Out** or **File** ► **Check Out All** to open the **Check Out** dialog box.
- 3 Optionally, check **Force Check Out** to overwrite any files with the same name in your working folder, even if they are more recent.
- 4 Select one of the following options in the **Reference By** group box for the files you wish to check out:
 - ◆ **Current Revision:** The most current (tip) revision.
 - ◆ **Label:** A specific file revision. The existing view and revision labels are listed in reverse chronological order based on the time at which they were created. The view labels precede the revision labels in the list.
 - ◆ **Promotion State:** A specific promotion state.
 - ◆ **As Of:** The revision that was the tip revision at the specified date and time. Click the **Date/Year** button to use the calendar, and specify the time by typing in the time or using the spin boxes.
- 5 Select a **Lock Status** option.
 - ◆ **Unlocked:** Releases your lock on the files after check-in.
 - ◆ **Exclusive:** Indicates that you intend to make further changes to the files.
 - ◆ **Non-exclusive:** Indicates that you are working on the files and may possibly make changes.
 - ◆ **Keep Current:** Retains the current lock status.
- 6 Optionally, click **Advanced** to open the **Advanced Options** dialog box.
 - ◆ Select **Default working file location**, or if you want to check the files out to a folder other than your designated working folder, select **Other** in the **Check Out Location** group box . If you select **Other**, browse for the folder name.
 - ◆ In the EOL Conversion area, click **None** or one of the other radio buttons to change your current EOL conversion setting for checking out text-based files. For Windows, the EOL marker is **CR-LF** (carriage return/line feed); for UNIX, it is **LF** (line feed); for Macintosh operating systems, it is **CR** (carriage return). EOL settings on this dialog override the default setting you selected on the **File** tab of the **Personal Options** dialog box.
 - ◆ Select an appropriate **File Encoding** from the drop-down list to support keyword expansion for non-English code pages..
 - ◆ Click **Close** to return to the **Check Out** dialog box.
- 7 Click **OK** to check out the files.

Note: You can also use the **Check Out** button on the toolbar to check out files without using the **Check Out** dialog box. If process rules are required, the **Check Out** dialog box will open automatically.

To check out a previous revision of a file

- 1 Click the **File** tab in the upper pane and select the file.
- 2 Click the **History** tab or the **Label** tab in the lower pane and select the revision to check out.
- 3 Right-click the selected file and choose **Check Out** or **Check Out To**.
This opens the **Check Out** dialog box.
- 4 Use the **Check Out** dialog box to check out the file as described in the procedure above for checking out files.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Adding Files to Projects

You can add files to a project folder if the files are in the correct working folder.

To add files to a StarTeam project folder

- 1 Put the files you want to add to the project into the local working folder that corresponds to the repository folder in StarTeam.
- 2 Open the project view in StarTeam and select the folder to which you want to add the files.
- 3 Select **Files Not In View** in the **Filter** list box on the toolbar,
This displays only the files in your working folder that have not been added to the project view.
- 4 Select the file(s) you want to add to the project, and choose **Files** ► **Add Files**.
- 5 Type a generic description for all the files in the **Add Files** dialog box, or check **Prompt for description for each file**.
Select an appropriate lock status for the files.

Note: The default is **Unlock**.

- 6 Check **Link and pin process item** to link the new files to process items, if process rules are enforced.
- 7 Check the **Mark selected process item as fixed/finished/complete** check box, if work on the active process item is final.
- 8 Check the **Show property editor for selected process item** check box to make changes to the selected process item's properties during the add process.
- 9 Optionally select a revision label from the drop-down list for the **Revision Label** field, or create a new revision label by typing the label name.
Existing labels list in reverse chronological order, based on the time at which they were created.
- 10 Click **OK** to add the files.

Note: You can also click **Advanced** to select advanced options, such as performing an EOL conversion and selecting a file encoding. EOL conversion is based on the **EOL** settings in the **Personal Options** dialog box.

When you add new files to the project, the status for these files changes from **Not In View** to **Current**.

If you check **Delete working files** during check in, the status of the new files changes to **Missing**.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Customizing Personal Options](#)
[Filtering Data](#)

Checking Out Files in Batches

Borland recommends that you use the Bulk Check-out (`bco`) utility when you need to check out files for builds because it is faster than the `stcmd co` command, especially when combined with the Cache Agent.

To check out files using the Bulk Check-out utility

- 1 In a Command window, type the `bco` command using the following syntax:

```
bco [options] [files...]
```

Where options can be any of the following

```
-autoLogon
-cfgl label | -cfgd date | -cfgp promotion state
-cmp
-csf
-cwf
-dryrun
-encrypt RC4 | RC2_ECB | RC2_CBC | RC2_CFB
-eol on | off | cr | lf | crlf
-exclude <pattern> | @ <pattern file>
-fp directory
-filter {CGIMOU}
-fs
-h[elp] or -?
-is
-netmon
-o
-p [user[:password]@]address[:port]/project[/view][folder/folder/...]
-pwdfile file
-rp directory
-ro
-t
-ts
-useca host:port | autolocate
-vb or -verbose
```

- 2 Use the following examples as a guideline for creating your `bco` command:

The following example uses `bco` to check out the files from [Source Code](#), a child of the root folder `StarDraw` (in the `StarDraw` view of the `StarDraw` project):

```
bco -p "JMarsh:password@Orion:1024/StarDraw/StarDraw/Source Code" -is -o ""
```

The next example shows a `bco` command that uses `-netmon` and the output displayed by Net Monitor.

```
bco -p "Administrator:Administrator@10.50.6.91:49201/StarDraw/WebSite" -fp D:\Test -netmon -o "*.htm"
```

Output:

```
StarTeam BulkCheckOut Utility version 11.0.2
Copyright (c) 2004-2009 Borland Software Corporation. All rights reserved.
```

```
Start: (rev 100) SRVR_CMD_GET_PROJECT_LIST Time: 62 millis; Sent: 42 bytes; Got: 1834 bytes
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 47 millis; Sent: 46 bytes; Got: 186 bytes
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 15 millis; Sent: 46 bytes; Got: 186 bytes
Start: (rev 100) SRVR_CMD_PROJECT_OPEN Time: 188 millis; Sent: 70 bytes; Got: 120 bytes
Start: (rev 100) PROJ_CMD_GET_VIEW_PROPERTIES Time: 31 millis; Sent: 42 bytes; Got: 2556
bytes
Start: (rev 100) PROJ_CMD_GET_FOLDERS Time: 63 millis; Sent: 42 bytes; Got: 1112 bytes
Start: (rev 100) PROJ_CMD_GET_FOLDER_ITEMS Time: 16 millis; Sent: 50 bytes; Got: 40 bytes
Start: (rev 100) PROJ_CMD_REFRESH_ITEMS Time: 3562 millis; Sent: 122 bytes; Got: 414 bytes
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 16 millis; Sent: 46 bytes; Got: 186 bytes
Start: (rev 100) PROJ_CMD_GET_PROJECT_PROPERTIES Time: 31 millis; Sent: 42 bytes; Got: 4797
bytes
Start: (rev 100) FILE_CMD_CHECKOUT Time: 47 millis; Sent: 78 bytes; Got: 108 bytes
Start: (rev 100) FILE_CMD_CHECKOUT Time: 31 millis; Sent: 78 bytes; Got: 1767 bytes
Start: (rev 100) FILE_CMD_CHECKOUT Time: 31 millis; Sent: 78 bytes; Got: 1140 bytes
Start: (rev 100) SRVR_CMD_PROJECT_CLOSE Time: 15 millis; Sent: 62 bytes; Got: 16 bytes
Start: (rev 100) SRVR_CMD_RELEASE_CLIENT Time: 31 millis; Sent: 42 bytes; Got: 16 bytes
```

Related Concepts

[Bulk Check-out Utility for Large Numbers of Files](#)

Checking Out Historical Versions of Files

You can easily check out a previous revision of a file using the **History** pane. You will have a choice whether to check out the file to the current working folder, which would overwrite the current file, or to check out to a different location.

To check out a previous revision of a file

- 1 Click the **File** tab in the upper pane and select the file.
- 2 Click the **History** tab or the **Label** tab in the lower pane, and select the revision to check out.
- 3 Right-click the selected file and choose **Check Out** or **Check Out To**.
This opens the **Check Out** dialog box.
- 4 Use the **Check Out** dialog box to check out the file as described in the procedure above for checking out files.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Comparing Historical File Contents](#)
[Comparing Historical Properties](#)
[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Controlling How File Status Information is Stored

File status information about the files you are working on is stored on your workstation either in a central location or in a child folder (named `.sbas`) of each working folder.

You can set the file status property for a specific view. The view property defaults to the storage method that you selected as a personal option. When changed from that default, the view property take precedence over your personal option for the view.

Note: You can also set your Personal Options to control file status information for all your files, unless those files are in views for which you have set the view property for file status.

To set the file status property for a view

- 1 Choose **View** ► **Properties** to open the **View Properties** dialog box, then select the **Name** page.
- 2 Select the **Central** or **Per Folder** option button in the **File Status Repository** group box.

The per-folder option is most useful in the special case where multiple users are sharing a working folder; for example, on a shared network drive.

For example, suppose several users all check files in and out of a shared working folder. If these users have set the central repository option for file statuses, the statuses are stored on each of their computers. Whenever a user makes a change to a file in the working folder, the status for that file is undated only on that user's computer. Everyone else sees the status **Unknown** for that file. Over time, all the files may have been changed, and the statuses can become **Unknown** for all users of all files. Using the per-folder option causes the statuses to be updated within the working folder itself. Everyone has access to those status changes and **Unknown** statuses do not occur.

- 3 Click **OK**.

Tip: Select **Default (Central)** to return to using the **Personal Options** settings.

Related Concepts

[Personal Options](#)
[Proper Use of Views](#)

Related Procedures

[Customizing Personal Options](#)
[Reviewing or Modifying View Properties](#)

Related Reference

[File Status Information](#)
[Effects of Status on Check-ins and Check-outs](#)

Deleting Folders and Items

The procedures below explain how to delete folders and items in the StarTeam client. They also explain how to delete the your working folders without deleting the folders in the view or project.

To delete a StarTeam folder

- 1 Select the folder you want to delete in the project view.
You can do this in the folder hierarchy tree on the left, or on the **Folder** tab in the upper pane.
- 2 Right-click the selected folder and choose **Delete**.
A dialog box opens prompting you to confirm the deletion.
- 3 Optionally, check **Delete Working Folders** to include your local folder and its contents.
- 4 Click **Yes** to delete the folder.

StarTeam deletes the selected folder and all its files and sub-folders.

To delete a local folder only

- 1 Select the folder you want to delete in the project view.
You can do this in the folder hierarchy tree on the left, or on the **Folder** tab in the upper pane.
- 2 Right-click the selected folder and choose **Delete Local Folders**.
A dialog box opens prompting you whether you really want to delete the selected folder and all its files and sub-folders.
- 3 Click **Yes**.

To delete an item

- 1 Click the tab in the upper pane containing the item or items you want to delete: **File**, **Change Request**, **Requirement**, **Task**, or **Topic**.
- 2 Select one or more items to delete.
- 3 Right-click the selected item(s) and choose **Delete**.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Displaying Item Details

You can view an item's details quickly using the **Detail** tab in the lower pane. The Detail pane displays the item properties and their values in a two-column list.

The properties that display in the **Detail** pane for file, change request, child folder, and audit items are determined by which fields you choose to display in the upper pane, and/or which filters you use for the component tab.

To display item details

- 1 Click a component tab in the upper pane.
- 2 Select an item in the upper pane.
- 3 Click the **Detail** tab in the lower pane.

Related Procedures

[Displaying Additional Fields](#)

[Applying Predefined Filters](#)

Editing Check-in Comments

Normally you enter comments during the check-in process describing the changes you are checking in. You can also add or edit check-in comments after files have been checked in.

To add or edit a check-comment after a file has been checked in

- 1 Select the file in the upper pane on the **File** tab, and click the **History** tab or the **Label** tab.
- 2 Select the revision for which you want to add or edit the comment in the **History** or **Label** pane.
- 3 Right-click the selected revision and choose **Edit Comment** to open the **Edit Comment** dialog box.
- 4 Type the text you want for the comment and click **OK**.

Comment fields allow up to 30,000 characters.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Enabling Concurrent File Editing

The personal option named **Use Non-Exclusive Locks In Integrations** affects how files are locked when accessed from application integrations such as Visual Studio. If you select this option, locking a file (for example, as part of a check-out operation) creates a non-exclusive lock rather than an exclusive lock.

With an exclusive lock, only the person who has the file locked can check in the file. With a non-exclusive lock, others can check in the file. Exclusive locks are the safest, but non-exclusive locks are often preferred because text files can be easily merged using File Compare/Merge. Using non-exclusive locks allows more than one person to edit a file at one time. If team members are not editing the same lines of the file, the merged file usually has no conflicts.

If you are using an application integration for your development environment (for example, the integration with Visual Studio, you cannot check in files from the development environment if both the **Require Exclusive Lock When Files Are Checked In** check box in the **Project Properties** dialog box, (**Options** tab) and the **Use Non-exclusive Locks In Integrations** check box in the **Personal Options** dialog box (**Files** tab) are checked. The administrator usually determines the setting of the **Require Exclusive Lock When Files Are Checked In** check box. However, personal options are set by you for your workstation.

To use non-exclusive locks with integrations

- 1 Choose **Tools** ► **Personal Options** to open the **Personal Options** dialog box.
- 2 Click the **Files** tab and check **Use Non-exclusive Locks In Integrations**.
- 3 Click **OK**.

Note: If you have checked **Use Non-exclusive Locks In Integrations** and experience check-in problems, try clearing the check box. You may want to talk to your administrator about the setting for the **Require Exclusive Lock When Files Are Checked In** check box.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Excluding Files and Folders from a Project

Some types of files will never be added to a project, although they may reside in a working folder. For example, suppose you are creating files with an application that makes an automatic backup (.bak) copy of each file every time you save the file. Although your working folder might contain several .bak files, you would have no reason to check them into (or out of) the application. Therefore, you should exclude them from the project view.

Exclude lists can also be inherited from parent folders.

To exclude files from a project

- 1 Select a folder from the application folder hierarchy tree, or from the **Folder** tab in the upper pane.

Note: You can also exclude files in Not-in-View folders, but you must have the root folder selected to do so.

- 2 Right-click the selected folder and choose **Properties** to open the **Folder Properties** dialog box.
- 3 Select the **Exclude** tab.

Note: The **Exclude** tab does not affect files that are already part of the project.

- 4 Choose one of the following:

- ◆ **Inherit and Use Local Exclude List:** Excludes files that match the exclude list specifications set for this folder as well as those of its parent folder. If the **Local Exclude List** text box does not yet include any file specifications, add them.
- ◆ **Use Local Exclude List:** Excludes files that match the exclude list specifications set for this folder. If the **Local Exclude List** text box does not yet include any file specifications, add them.
- ◆ **No Exclude List:** includes all files.

- 5 Type one or more file specifications to use for matching files.

Use the standard expressions (with * and ? wild cards) separated by commas, spaces, or semicolons. To include a comma, space or semicolon as part of the specification, enclose the specification in double quotes.

A trailing / character represents Not-in-View folders to be excluded. For example, bin/ would cause all Not-in-View folders named bin to be excluded from the folder tree.

Note: The \ character does not work; it is treated as an escape character.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Finding Files Linked to Active Process Items

When you have files linked to an active process item, you can quickly find all the files linked to it.

To find files linked to an active process item

- 1 Click the **Change Request**, **Requirement**, or **Task** tab in the upper pane that contains the active process item.
You can see what item is the active process item by looking at the left side of the **Status Bar**. The second box in the **Status Bar** displays the **Active Process Item** icon, followed by the name of the item.
- 2 Select the active process item and click the **Link** tab in the lower pane.

The **Link** pane displays all the files linked to that item.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)

Hiding Folders and Files

Using the **Folder Properties** dialog box, you can set the **Visible** property to exclude folders and their files from visibility.

To hide a folder

- 1 Do one of the following to open the **Folder Properties** dialog box:
 - ◆ Double-click the folder on the **Folder** tab.
 - ◆ Right-click the folder in the **Folder Tree** and choose **Properties**.
- 2 Select the **Name** tab In the **Folder Properties** dialog box and uncheck the **Visible** option to hide the folder and the files it contains.
- 3 Click **OK** to close the dialog box and save the folder properties.

Note: To make the folder visible again, check the **Visible** option in the **Folder Properties** dialog box.

Related Concepts

[Overview of Folders and Paths](#)

[Files](#)

[Folders](#)

Related Procedures

[Excluding Files and Folders from a Project](#)

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Locking and Unlocking Items

Before changing the contents of a file or editing item properties, you should exclusively lock the file or item. This action informs other team members that you intend to make changes. Files, change requests, requirements, tasks, and topics can all be locked.

Exclusively locking an item prevents others from creating new revisions of it before the lock has been released. You can lock and unlock any type of item as a separate operation. In addition, you can lock and unlock files as part of the check-in and check-out processes.

If an item is exclusively locked by someone else, you can review its properties but cannot change them. Normally the words **Read Only** and the name of the user who has locked the item will appear on the title bar.

To lock an item using the toolbar

- 1 Select one or more items in the upper pane.
- 2 Click the **Lock** button on the toolbar.

The selected items become exclusively locked, and you are listed as the user who has locked them.

To lock an item using a menu

- 1 Select a folder in the folder hierarchy tree on the left.
- 2 Click any component tab in the upper pane except **Audit**. (Audit items cannot be locked.)
- 3 Right-click an item in the upper pane and choose **Lock/Unlock**.
This opens the **Set My Lock Status** dialog box.
- 4 Select a lock status option:
 - ◆ **Unlocked**: Removes your exclusive or non-exclusive lock on the selected items
 - ◆ **Exclusive**: Prevents others from creating new revision of this item (until you release the lock or another person breaks your lock).
 - ◆ **Non-exclusive**: Indicates that you are working on the item and may possibly make changes (not recommended for items other than files).
- 5 Optionally, check **Break Existing Lock** to break another team member's lock on the item.
If e-mail is enabled, StarTeam will send an e-mail message to the team member whose lock has been broken to inform him or her of this fact.

Note: You must be granted the appropriate privileges to be able to break another person's locks.

To remove your lock from an item

- 1 Select the locked item.
- 2 Click the **Unlock** button on the toolbar

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Marking Unlocked Files Read-only

In many cases, users make edits before realizing that their files must be exclusively or non-exclusively locked to check them in. If the files are read-only, users are less likely to make this mistake.

To make unlocked files read-only

- 1 Select **Project** ► **Properties** from the menu bar.

The **Project Properties** dialog opens.

- 2 Select the **Options** tab.

- 3 Check **Mark Unlocked Working Files Read-only**, which applies to files that are unlocked in the application or in application integrations with third-party applications.

If this check box is cleared, you must use the operating system to change the read-only attribute to read/write.

Working copies of unlocked files will now become read-only when the following file operations are performed:

- ◆ File check-ins
- ◆ File check-outs (from the **File** or **History** pane)
- ◆ File unlocks

Note: This project property overrides the identical **Mark Unlocked Working Files Read-only** personal option. If you change your mind after selecting the property (or the equivalent personal option), verify that no files are writable before clearing the check box. Next, force a check out and lock all the files (or just the read-only files). Finally, unlock them.

Related Concepts

[Overview of Projects](#)
[Process Items Overview](#)

Related Procedures

[Creating Projects](#)
[Viewing or Modifying Project Properties](#)
[Changing Project Names or Descriptions](#)
[Requiring Revision Comments](#)
[Requiring Exclusive Locks for Check-ins](#)

Monitoring Check-out Statistics using Cache Agent

The visible advantage to using Cache Agent is the improved speed of file check-out operations. The more files you check out, the more advantage you will gain from Cache Agent. Over time, more and more of the files will come from Cache Agent, reducing the strain on StarTeam Server. As a result, the check-out speed should continue to improve until all files are available from Cache Agent.

For a particular check-out operation, you can see how many files are being sent by StarTeam Server directly and how many are being sent by Cache Agent, by displaying the check-out statistics.

To monitor check-out statistics using Cache Agent

- 1 Select the files to be checked out In the StarTeam client.
- 2 Select **File** ► **Check Out** from the main menu. The **Check Out** dialog box opens.
- 3 Check **Show check-out statistics**.
- 4 Select any other option settings that are appropriate to your check-out operation, and click **OK**.
After the check-out operation completes, the **check-out Statistics** dialog box opens displaying the elapsed time, total number of files, total number of bytes, and whether any of the files were failed or skipped during the check-out.
- 5 Click **OK** to close the dialog box.

Note: The option to **Show check-out statistics** is also available in some of the StarTeam IDE-based integrations too.

Opening a Local Folder from StarTeam

The following procedures describe how to quickly open a local folder in a file browser from within the StarTeam client so you can perform basic file and folder management tasks.

To open a local folder from a folder selection in StarTeam

- 1 In StarTeam, select from one to five folders you want to open.
You can select one folder in the folder hierarchy tree on the left or up to five folders on the **Folder** tab in the upper pane.
- 2 Right-click the selected folder and choose **Open Local Folder**.

This opens a **Windows Explorer** for each location on disk that corresponds to a selected folder. This applies to all folders except those whose status is **Missing** since their local folders do not exist.

To open a local folder from a file selected in StarTeam

- 1 On the **File** tab in the upper pane, select one to five files.
- 2 Choose **File** ► **Open Containing Folder**.
This opens a **Windows Explorer** to each folder in which a selected file exists. This applies to all folders except those whose status is **Missing** since their local folders do not exist.

Related Concepts

[Folders](#)

[Files](#)

Related Procedures

[Working with Folders and Items](#)

Opening and Editing Files

You can open a file directly from within StarTeam if the file has a status of **Current**. You can also display a file in the default editor, Notepad, or an alternate editor. If the file is an executable, such as a `.bat` file, or has no associated application, you must use the **Edit** command to edit the file from the application.

When you open a file in StarTeam, the application does one of the following:

- ◆ Runs the file if it is an executable, such as `autoexec.bat`.
- ◆ Displays an error messages if the file is not an executable and there is no associated application.
- ◆ Opens the file in an associated application. For example, `.doc` files will open in Microsoft Word for Windows. However, this only works if an “open” action file association exists for the selected file’s extension in the operating system.

To open a file from within StarTeam,

- 1 Click the **File** tab in the upper pane.
- 2 Choose **File** ► **Open**, or double-click the file.

If the file does not open in an associated application, an association may not have been created for the selected file type. See your operating system documentation for instructions on associating file types with applications.

To edit a file in StarTeam

- 1 Click the **File** tab in the upper pane.
- 2 Choose **File** ► **Edit**, or right-click the file and choose **Edit**.

Note: The file opens in the default editor, which is Notepad, unless you set an alternate editor as described below.

To change the default editor in StarTeam

- 1 Choose **Tools** ► **Personal Options** and click the **File** tab.
- 2 Click the **Alternate Applications** button to open the **Alternate Applications** dialog box.
- 3 Check **Editor** and browse to the executable file for the editor you want as the default application.
- 4 Type any appropriate **Options** and click **OK**.

Note: For non-Windows systems, specify a command to use for launching files in an alternate application in the **Open with...** text box on the **File** or **Folder** tab in the **Personal Options** dialog box.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Related Reference

[File Options \(Personal Options Dialog Box\)](#)

[Folder Options \(Personal Options Dialog Box\)](#)

Renaming Files

If you want to rename a file in your project, you should rename it within the StarTeam application. This retains the properties associated with that file, such as history and links. However, when you roll a view or item back to a configuration prior to the name change, the old name is not restored. This can cause problems. For example a setup script from an earlier build or release might fail because it uses the file's former name.

If you rename a file outside the application (for example, by using Windows Explorer), the application considers the file to be a new file. When you add the file with the new name, it will have no connection to the history, links, or other properties of the original file.

To rename a file

- 1 Select the file you want to rename in the upper pane.
- 2 Do one of the following:
 - ◆ Choose **File** ▸ **Properties**.
 - ◆ Right-click the selected file and choose **Properties**.
 - ◆ Click the **Properties** toolbar button.
- 3 Click the **General** tab in the **File Properties** dialog box.
- 4 Change the **Name** and click **OK**.

StarTeam renames the file in both the StarTeam repository and in your working folder.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Selecting Linked Files

You can quickly select all the files associated with a linked item directly from the component tab in the upper pane. There are two choices for selecting the linked files:

- ◆ You can select only the files linked to an item.
- ◆ You can add the files linked an item to an existing selection of files on the **File** pane.

To select all the files linked to a specific item

- 1 Click the component tab in the upper pane containing the item to which the files are linked, and select the item.
- 2 Right-click the selected item and choose **Linked Files** ► **Add To Selection**.

StarTeam switches the upper pane to the **File** pane, activates the **All Descendants View**, and adds all the files linked to the item to the existing file selection.

Note: Selected items must have linked files in the current view to perform this operation.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Customizing Link Properties](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Setting File Storage Options

StarTeam stores status information for the files in your working folder in a central location on your workstation or in a child folder (named `.sbas`) of each working folder. You can control how and where that information is stored by:

- ◆ Setting file status storage for a specific view. Normally, this view property defaults to the storage method that you have selected as a personal option. You can, however, select a storage method for a specific view. If you do so, this setting takes precedence over the personal option.
- ◆ Setting file status storage for all your files through a personal option setting that controls file status information for all your files. This setting applies to all files that are in views for which you have not set a file status property.

To set file status storage for a specific view

- 1 Choose **View** ► **Properties** to display the **View Properties** dialog box.
- 2 Select **Central** or **Per Folder** in the **File Status Repository** group box.

With the **Central** repository option, if several users all check files in and out of StarTeam from a shared working folder, file statuses are stored on each of their computers. Whenever a user makes a change to a file in the working folder, the status for that file is updated only on that user's computer. Everyone else sees the status **Unknown** for that file. Over time, all the files may have been changed, and the statuses can become **Unknown** for all users of all files.

Using the **Per Folder** option is most useful in the case where multiple users are sharing a working folder because it causes the statuses to be updated within the working folder itself. Everyone has access to those status changes and **Unknown** statuses do not occur.

- 3 Click **OK**.

To set file status storage for all your files

- 1 Choose **Tools** ► **Personal Options** and click the **File** tab.
- 2 Select the **Central** or **Per Folder** option in the **File Status Repository** group box.
- 3 Click **OK**.

Related Concepts

[Personal Options](#)

[Check-in and Check-out Overview](#)

Related Procedures

[Customizing Personal Options](#)

Related Reference

[File Status Information](#)

[Effects of Status on Check-ins and Check-outs](#)

[File Options \(Personal Options Dialog Box\)](#)

Setting the File Executable Bit for UNIX

When you add a file from a UNIX operating system, the state of the executable bit is preserved by StarTeam. For each file, there is an **Executable** check box that becomes selected if the executable bit is set and becomes cleared if the bit is not set. Future check-out operations ensure that the executable bit for the checked-out file matches the setting of the **Executable** check box.

To change the state of the executable bit

- 1 Select a file in the upper pane.
- 2 Choose **File** ► **Properties**.
- 3 On the **General** page of the **File Properties** dialog box, check or uncheck **Executable**.
- 4 Click **OK** to close the dialog box and apply the setting.

Related Concepts

[Overview of Folders and Paths](#)

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Using the File Annotation Tool

The File Annotation tool shows historical information about changes made to any text file in StarTeam.


To use the File Annotation tool

- 1 Choose **Tools** ► **File Annotation** from the main menu to enable the File Annotation tool.
The File Annotation tool will be enabled for every view window opened in the current Client session.
- 2 Select a text file from the **File** tab to view the tip revision's complete annotation information, or from the **History** tab to view changes made to a specific revision of the file.

Note: Annotation information can also be viewed from the **Link** tab if the item is a text file.

Like the File Compare tool, the File Annotation tool displays the contents of the selected file in an embedded pane at the bottom of the StarTeam window. If you resize the File Annotation pane, the Client will automatically display the split pane bars in the same position in every open view window.


Line by line, annotation information shown in this initial view of the file includes:


- ◆ The name of the user who made each change.
- ◆ A link to each process item linked to the revision.
- ◆ The dot notation of the file revision in which each change was made, which is linked to a view of the actual changes that were made to the content in that revision.
- ◆ An  **Info** icon that provides more details about the file revision and linked process items, such as the modified time or any comments.


To review the specific lines that were added, edited, or deleted in a previous version of the selected file


- 1 Click the dot notation link of the file revision you wish to review.
A new page will open that shows, line by line, the changes that occurred in the selected revision of the file.
- 2 Use the icons described in the table below to navigate the changes made to each version of the selected file you review.


File Annotation Icons

 Click this icon to go to the previous page.


 Click this icon to go to the next page.


 Click this icon to go to the end of the file.

 Click this icon to go to the beginning of the file


 Click this icon to go to the first changed line in the file.


 Click this icon to go to the last changed line in the file.

 Click this icon to go to the next changed line in the file.

 Click this icon to go to the previous changed line in the file.

 This icon denotes a line that has been added to the file. The text of added lines is green.

 This icon denotes each line that has been revised. The text of revised lines is blue, with additions highlighted in green and deletions highlighted in red with a strikethrough effect.

 This icon denotes each line that has been deleted from the file. The text of deleted lines is red with a strikethrough effect.

Viewing or Modifying Item Properties

This section explains how to use the standard properties dialog to edit item properties. Depending on how your team has set up the application, you may see a totally different dialog box called an alternate property editor (APE).

Every time the properties of an item are modified, a new revision of that item is created. If you modify a property, you should also create a revision comment explaining the modification using the **Revision Comment** tab.

To view or modify item properties

- 1 Click a folder in the folder hierarchy tree, or click one of the component tabs in the upper pane and select an item.
- 2 Do one of the following:
 - ◆ Right-click the folder or item and choose **Properties**.
 - ◆ Right-click an item on the **Reference** tab in the lower pane and choose **Item Properties**.
 - ◆ Click the **Properties** toolbar button.
- 3 Modify any of the property fields in the corresponding **Properties** dialog box that opens, then click **OK**.

Tip: You can force Java to not cache Windows file properties by placing the following option in the CPC stjava file:

```
-Dsun.io.useCanonCaches=false.
```

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)
[Viewing or Modifying Item Properties](#)

Related Reference

[Change Request Properties](#)
[File Properties](#)
[Folder Properties](#)
[Requirement Properties](#)
[Task Properties](#)
[Topic Properties](#)

Viewing Previous File Revisions

You can review the contents of a prior file revision in either the default editor or in the application for which the file type is registered.

To review a revision's contents

- 1 On the **File** tab, select the file for which you want to review a revision.
- 2 On the **History** or **Label** tab in the lower pane, select the specific revision you want to review.
- 3 Right-click the selected item to open the context menu and choose one of the following:
 - ◆ **View Revision Content** to copy the revision to a temporary file and display it in the default editor (Notepad or the alternate editor specified in the **Personal Options** dialog box).
 - ◆ **Open Revision Content** to copy the revision to a temporary file and display it in the associated application.

Note: The client creates the temporary files in the local temp directory on the system. For example, if working on a Windows system, the temporary files are created in the `C:\Documents and Settings\<user>\Local Settings\Temp` directory. When you exit the client, the files are deleted from the system.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Working with Change Requests

This section contains tasks related to working with change requests.

In This Section

[Adding Change Request Comments](#)

Describes how to add comments to change requests.

[Assigning Change Requests](#)

Describes how to assign change requests.

[Closing Verified Change Requests](#)

Describes how to close verified change requests.

[Creating Change Requests](#)

Describes how to create change requests.

[Customizing Change Request Filters](#)

Describes creating a change request filter based on the current arrangement of data.

[Customizing Change Request Reports](#)

Describes how to customize a change request report.

[Including Change Request Attachments](#)

Describes how to include file attachments with change requests.

[Linking Items Internally or Externally](#)

Describes how to link folders and items both internally in the same view, or externally between items on two different servers or configurations.

[Locking and Unlocking Items](#)

Describes how to lock and unlock items.

[Modifying Custom Options for Change Requests](#)

Describes how to modify custom options for change requests.

[Resolve Open Change Requests](#)

Describes how to resolve open change requests.

[Reviewing Linked Change Requests](#)

Describes how to review linked change requests.

[Selecting Change Requests Using a Query](#)

Describes how to select change requests using a query.

[Showing Fields in a Change Request](#)

Describes how to show specific fields in a change request.

[Sorting and Grouping Change Requests](#)

Describes how to sort and group change requests.

[Specifying Change Request Descriptions](#)

Describes how to specify change request descriptions.

[Specifying Change Request Solutions](#)

Describes how to specify change request solutions.

[Specifying Change Request Summary Information](#)

Describes how to specify change request summary information.

[Verifying Resolved Change Requests](#)

Describes how to verify a resolved change request.

[Working with Change Request Lists](#)

Describes how to work with change request lists.

Adding Change Request Comments

You can add comments to a change request, such as the reason for changing the change request properties.

To add comments to a change request

- 1 Click the **Comments** tab.

Note: The Comment for this revision lists any comments that were entered for the current version of the change request only. That is, each time you change the change request and type a comment, the new comment replaces the old comment when you save the change request.

- 2 Type your comments in the **Comment for new revision** text box.

Note: You must make a change to a property of this change request before you can type a comment.

- 3 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Descriptions](#)

[Specifying Change Request Solutions](#)

[Modifying Custom Options for Change Requests](#)

[Including Change Request Attachments](#)

Related Reference

[Change Requests](#)

Assigning Change Requests

Assigning a change request refers to assigning the status of the change request as well as who is currently responsible for the change request.

To assign change requests

- 1 Select the appropriate folder and click the **Change Requests** tab from the upper pane.

Tip: To find all change requests in a folder, select the **All Descendants** button on the toolbar. If desired, you can create a filter or query to find all change requests with **New** status or simply sort the **Status** column in the upper pane to find all **New** change requests.

- 2 Select the change request, then choose **Properties** from the **Change Request** menu or context menu. The **Change Request Properties** dialog box opens.
- 3 Review the settings and decide on an appropriate status. You can select **Open**, **Is Duplicate**, **As Designed**, or **Deferred**.
 - ◆ If you select **Open**, the **Responsibility** changes to the person best qualified to fix or enhance the product, as described in the change request.
 - ◆ If you select **Is Duplicate** or **As Designed**, the **Responsibility** changes to the person who submitted the change request. The assumption is that the person who submitted the change request will want to know about, verify, or perhaps challenge this change in status.
- 4 Click **Apply**, then click the **Next** or **Previous** button to review another change request.

Related Concepts

[Change Requests](#)

[User Roles and StarTeam Documentation](#)

Related Procedures

[Creating Change Requests](#)

Related Reference

[Change Requests](#)

Closing Verified Change Requests

To close verified change requests

- 1 Select the appropriate folder and click the **Change Requests** tab from the upper pane.
- 2 Optionally, expand or limit the number of change requests that are displayed by doing the following:
 - ◆ Select the **All Descendants** button on the toolbar.
 - ◆ Create a filter or query to display the appropriate change requests or simply sort the **Status** column in the upper pane to group the change requests.
- 3 Double-click the change request you want to close. The **Change Request <number, revision #>** dialog box opens.
- 4 Change the status to **Closed**. StarTeam has the following “closed” statuses:
 - Closed (As Designed)**
 - Closed (Cannot Reproduce)**
 - Closed (Deferred)**
 - Closed (Documented)**
 - Closed (Fixed)**
 - Closed (Is Duplicate)**
- 5 Do one of the following:
 - ◆ Click **Apply**, then click the **Next** or **Previous** button to close another change request.
 - ◆ Click **OK** to close this dialog.

Related Concepts

[Change Requests](#)

Related Procedures

[Resolve Open Change Requests](#)

[Verifying Resolved Change Requests](#)

Related Reference

[Change Requests](#)

Creating Change Requests

Development teams create change requests to record problems and enhancement requests and track their resolution or implementation. Many teams have adopted specific guidelines that govern the creation and content of change requests – for example, instructions about what can be entered in the **Component** and **Category** fields. Be sure to follow these guidelines, if they exist.

To create a change request

- 1 Define the change request summary information on the **Synopsis** tab.
[Specifying Change Request Summary Information](#)
- 2 Describe the change request on the **Description** tab.
[Specifying Change Request Descriptions](#)
- 3 Optionally, specify a temporary workaround on the **Solution** tab.
[Specifying Change Request Solutions](#)
- 4 Optionally, specify custom change request properties on the **Custom** tab.
[Modifying Custom Options for Change Requests](#)
- 5 Optionally, attach files related to this change request on the **Attachments** tab.
[Including Change Request Attachments](#)
- 6 Optionally, add comments about the change request on the **Comments** tab.
[Adding Change Request Comments](#)
- 7 Click **OK** to save the change request. StarTeam assigns a unique number to the change request and displays the summary information in the upper pane.

Related Concepts

[Change Requests](#)
[User Roles and StarTeam Documentation](#)

Related Reference

[Change Requests](#)

Specifying Change Request Summary Information

You use the **Synopsis** tab to define and modify the summary information about a change request. Summary information includes important criteria like status, severity, and who is currently responsible for this change request.

To specify the summary information for a change request

- 1 On the **Synopsis** tab, accept the default status **New** or select another status from the **Status** drop-down list box.
- 2 Indicate the severity of the change request by selecting **High**, **Medium**, or **Low** from the **Severity** drop-down list box.

Note: The team leader usually sets the criteria for high, medium and low status.

- 3 If the change request needs immediate attention, select **Yes** from the **Priority** list box.
- 4 To specify the type of change request, select **Defect** or **Suggestion** from the **Type** list box.
- 5 Select the platform to which the change request applies from the **Platform** drop-down list box.
- 6 Type a summary of the change request in the **Synopsis** text box. The application can accept a maximum of 20K characters in this text box, but your database may accept fewer characters.
- 7 Select the name of the team member responsible for correcting the change request from the **Responsibility** drop-down list box.
- 8 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Descriptions](#)
[Specifying Change Request Solutions](#)
[Modifying Custom Options for Change Requests](#)
[Including Change Request Attachments](#)
[Adding Change Request Comments](#)

Related Reference

[Change Requests](#)

Specifying Change Request Descriptions

You use the **Description** tab to specify detailed information about the change request including the steps to reproduce the problem.

To specify a description for a change request

- 1 Click the **Description** tab.
- 2 Type a detailed description of the change request in the **Description and steps to reproduce** text box.
Include the steps to reproduce the problem, or in the case of an enhancement request, a detailed description of the enhancement.
- 3 Optionally, type or browse for the path to a test for the change request in the **Test command** text box.
- 4 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Solutions](#)

[Modifying Custom Options for Change Requests](#)

[Including Change Request Attachments](#)

[Adding Change Request Comments](#)

Related Reference

[Change Requests](#)

Specifying Change Request Solutions

You use the **Solution** tab to specify a workaround for the problem and to document how this change request was resolved.

To specify a solution for a change request

- 1 Click the **Solution** tab.
- 2 Optionally, in the **Work around** text box, type the steps you can follow to work around the problem.
- 3 Optionally, in the **Fix** text box, type the solution to the problem. The **Fix** text box is usually completed by the user who fixes the code. In this box, the application can accept a maximum of 20K characters, but your database may accept fewer characters.
- 4 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Descriptions](#)

[Modifying Custom Options for Change Requests](#)

[Including Change Request Attachments](#)

[Adding Change Request Comments](#)

Related Reference

[Change Requests](#)

Modifying Custom Options for Change Requests

Your team leader may have created additional change request properties. You use the **Custom** tab to change the default properties.

To modify custom options for a change request

- 1 Click the **Custom** tab.
- 2 Double-click the name of the custom property. The **Edit Property** dialog box opens.
- 3 Select a new value for the property. For integer, text, and real fields, **Value** is a text box. For enumerated types and user IDs, it is a list box. For dates and times, **Value** has a **Date** check box and a **Time** check box, each of which is followed by a date or time in the format for your locale.
- 4 Repeat steps 2 and 3 for other custom properties.
- 5 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Descriptions](#)

[Specifying Change Request Solutions](#)

[Including Change Request Attachments](#)

[Adding Change Request Comments](#)

Related Reference

[Change Requests](#)

Including Change Request Attachments

To include file attachments with a change request

- 1 Click the **Custom** tab.
- 2 Click **Add** to attach a file to the change request.
- 3 From the **Open** dialog box, select the files to be attached.
- 4 Click **Open**.
- 5 Optionally, add more attachments.
- 6 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Descriptions](#)

[Specifying Change Request Solutions](#)

[Modifying Custom Options for Change Requests](#)

Related Reference

[Change Requests](#)

Adding Change Request Comments

You can add comments to a change request, such as the reason for changing the change request properties.

To add comments to a change request

- 1 Click the **Comments** tab.

Note: The Comment for this revision lists any comments that were entered for the current version of the change request only. That is, each time you change the change request and type a comment, the new comment replaces the old comment when you save the change request.

- 2 Type your comments in the **Comment for new revision** text box.

Note: You must make a change to a property of this change request before you can type a comment.

- 3 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Descriptions](#)

[Specifying Change Request Solutions](#)

[Modifying Custom Options for Change Requests](#)

[Including Change Request Attachments](#)

Related Reference

[Change Requests](#)

Customizing Change Request Filters

After you have sorted, grouped, selected columns, applied queries to the change requests component in the upper pane, you can save the arrangement of change request data that appears in the upper pane as a filter. You can later apply the filter to any change request data to view the data using the same arrangement.

To create a filter for the current change request arrangement

- 1 Right-click the column headers and select **Save Current Settings** from the context menu. The **Save Current Settings** dialog appears.
- 2 Type a filter name in the **Filter Name** text box.
- 3 Select or clear the **Public** check box depending on whether this filter is to be used by all or only on your workstation.
- 4 Click **OK**. The filter name will appear from now on in the **Filter** drop-down list box.

Related Concepts

[Filters](#)

[Change Requests](#)

Related Procedures

[Creating Filters](#)

[Applying Predefined Filters](#)

[Working with Change Requests](#)

[Customizing Change Request Reports](#)

Related Reference

[Change Request Filters](#)

Customizing Change Request Reports

You can create a number of reports using the change request report features. This topic demonstrates an example of creating a report of change requests fixed during a certain time period.

To create a report of change requests fixed during a certain time period

- 1 Select the root folder in the **Folder Tree**.
- 2 Click the **Change Request** tab.
- 3 Choose **Change Request** ► **All Descendants**.
- 4 On the **Change Request** tab, display the **CR Number**, **Status**, **Modified Time**, and **Modified By** fields.
[Showing Fields in a Change Request](#)
- 5 Define the query that includes these fields and specifies a date range.
Specify a beginning **Modified Time**, and if the end date is not the current date, use an AND operator and specify an ending **Modified Time**.
[Selecting Change Requests Using a Query](#)
- 6 Sort and group the change requests, selecting **Status** from the **First By** drop-down list box and checking **Group By**.
[Sorting and Grouping Change Requests](#)
- 7 Select the **Status:Fixed** group.
- 8 Choose **Change Request** ► **Reports** to create a report showing the fixed change requests.

Related Concepts

[Change Requests](#)

Related Procedures

[Working with Change Requests](#)
[Customizing Change Request Filters](#)
[Filtering Data](#)

Related Reference

[Fields](#)
[Change Requests](#)

Showing Fields in a Change Request

You can select which fields are displayed for a change request.

To show specific fields in a change request

- 1 Right-click the column headers and select **Show Fields** from the context menu.
This opens the **Show Fields** dialog box.
- 2 Make sure the **CR Number**, **Entered By**, and any other appropriate fields are displayed in the **Show these fields in this order** list.
- 3 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Selecting Change Requests Using a Query

You can use a simple or complex query to limit the change requests displayed to those that fit specific criteria.

To select change requests using a query

- 1 Click **Change Request** tab in the upper pane.
- 2 Right-click a column header and choose **Queries** from the context menu.
- 3 Choose an existing query in the **Queries** dialog.
- 4 Click **Select** to apply it to the list of change requests.
- 5 Follow the steps in “Creating Queries” to define a new query.

Related Concepts

[Change Requests](#)

[Queries](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Sorting and Grouping Change Requests

You can sort change requests by the data in a particular column or group the change requests.

To sort or group change requests

- 1 Click on a column header on the Change Requests tab to sort the change requests by the data in that column.

Note: If you want to sort or group the change requests first by the data in one column and then by the data in another column, see the following steps. You can sort or group the change requests in up to four levels of groupings.

- 2 To sort or group the change requests in multiple levels, right-click the column headers and select **Sort and Group** from the context menu. The **Sort and Group** dialog box opens.
- 3 From the **First By** drop-down list box, select a column title.
- 4 Optionally, group the change requests by the data in this column, select the **Group By** check box.

If you select the **Group By** check box, the change requests are grouped together in nested lists and you must drill down to view the change requests in each group. If you do not select the **Group By** check box, the change requests are all displayed on the **Change Requests** tab, sorted by your choices in **Sort and Group** dialog box.

Note: By default, the column data is sorted or grouped based on the internal key or order. You can use the **Sort Options** button and choose to sort or group the data based on the text and optionally, case sensitivity.

- 5 Repeat steps three and four to define up to four levels of sort orders or groupings.
- 6 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Including Change Request Attachments

To include file attachments with a change request

- 1 Click the **Custom** tab.
- 2 Click **Add** to attach a file to the change request.
- 3 From the **Open** dialog box, select the files to be attached.
- 4 Click **Open**.
- 5 Optionally, add more attachments.
- 6 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Descriptions](#)

[Specifying Change Request Solutions](#)

[Modifying Custom Options for Change Requests](#)

Related Reference

[Change Requests](#)

Linking Items Internally or Externally

This procedure describes how to link two items, either internally in the same server configuration, or linking between two items located on different server configurations, called *external linking*.

In StarTeam, an *item* is a file, change request, requirement, task, or topic. A *link* is a connection between two folders, two items, or a folder and an item. Creating links can be quite useful. For example, linking a file to a change request allows you to mark it as fixed when you check in the edited file. By linking files to the requirements document that the files fulfill, you can easily refer to or update the document.

You can create several links at the same time if you want to link several items of the same type to one particular item. For example, you might wish to link several change requests to a single file. To accomplish this, you can create links using the **Folder Tree** menu, component menu, context menu, or **Link** button on the toolbar.

Note: When you create external links between items on different server configurations, both server configurations need to be opened in the Cross-Platform Client to be able to create or view the external links.

To link a folder or item(s) to one or more items

- 1 Begin the link process by doing one of the following:
 - ◆ Open the server configuration and view which contains the items you want to link. If you want to link two items on different server configurations, open both server configurations and views in the Cross-Platform Client.
 - ◆ Select a folder in the folder hierarchy or in the upper pane on the **Folder** tab.
 - ◆ Click a component tab in the upper pane, such as **File**, **Change Request**, **Requirement**, **Topic**, or **Task**, and select one or more items.

Note: You cannot link two folders to each other with the method described in this procedure.

- 2 Right-click the selected item(s) on the component tab and choose **Links** ► **Create Link**.
The **Links** menu is also available on the component menu that corresponds with the selected component tab.
This action changes the mouse pointer and displays it as a knotted rope.

Note: If you initially select an item from the upper pane, you can also use the **Link** button on the toolbar; however, this button is disabled if you start the link with a folder.

- 3 Select the folder or item(s) for the end of the link from the same configuration or from the external server configuration. This can be:
 - ◆ A StarTeam folder (if you have not already selected a folder).
 - ◆ One or more other files.
 - ◆ One or more change requests.
 - ◆ One or more requirements.
 - ◆ One or more topics and/or responses.
 - ◆ One or more tasks and/or subtasks.

To locate all items, you may need to switch to a different component tab or use the **All Descendants** button on the toolbar.

- 4 Choose **Links** ► **Complete Links** on the **Folder** menu, the component menu, or the context menu, or click the **Link** button again on the toolbar.

This button is disabled if you are linking an item to a folder.

- 5 Verify that the links exist by doing one of the following:

- ◆ Select a linked item, then click the **Link** tab on the lower pane to view the links for the item.
- ◆ Right-click a linked folder, then choose **Properties** to display the **Folder Properties** dialog box. Click the **Link** tab to view the link. (The **Link** tab will not appear in this dialog box if you do not have access rights to view links.)

You can also view a link by selecting either of its ends. The end you select, whether a folder or an item, is called the source. The other end of the link is called the target and is listed in the **Item Type** column on the **Link** pane.

Tip: If you change your mind about creating a link after you have started to create it, but before you have finished completing it, you can select **Links** ► **Cancel Link** on the **Folder Tree** menu, the component menu, or the context menu. If you are using the **Link** button on the toolbar, press **ESC**.

Related Concepts

[Links: Internal and External](#)

Related Procedures

[Linking Specific Revisions](#)

[Checking Linked Files In and Out](#)

[Selecting Linked Files](#)

[Deleting Links](#)

[Linking Files to Process Items](#)

[Creating External Links](#)

Locking and Unlocking Items

Before changing the contents of a file or editing item properties, you should exclusively lock the file or item. This action informs other team members that you intend to make changes. Files, change requests, requirements, tasks, and topics can all be locked.

Exclusively locking an item prevents others from creating new revisions of it before the lock has been released. You can lock and unlock any type of item as a separate operation. In addition, you can lock and unlock files as part of the check-in and check-out processes.

If an item is exclusively locked by someone else, you can review its properties but cannot change them. Normally the words **Read Only** and the name of the user who has locked the item will appear on the title bar.

To lock an item using the toolbar

- 1 Select one or more items in the upper pane.
- 2 Click the **Lock** button on the toolbar.

The selected items become exclusively locked, and you are listed as the user who has locked them.

To lock an item using a menu

- 1 Select a folder in the folder hierarchy tree on the left.
- 2 Click any component tab in the upper pane except **Audit**. (Audit items cannot be locked.)
- 3 Right-click an item in the upper pane and choose **Lock/Unlock**.
This opens the **Set My Lock Status** dialog box.
- 4 Select a lock status option:
 - ◆ **Unlocked**: Removes your exclusive or non-exclusive lock on the selected items
 - ◆ **Exclusive**: Prevents others from creating new revision of this item (until you release the lock or another person breaks your lock).
 - ◆ **Non-exclusive**: Indicates that you are working on the item and may possibly make changes (not recommended for items other than files).
- 5 Optionally, check **Break Existing Lock** to break another team member's lock on the item.
If e-mail is enabled, StarTeam will send an e-mail message to the team member whose lock has been broken to inform him or her of this fact.

Note: You must be granted the appropriate privileges to be able to break another person's locks.

To remove your lock from an item

- 1 Select the locked item.
- 2 Click the **Unlock** button on the toolbar

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Modifying Custom Options for Change Requests

Your team leader may have created additional change request properties. You use the **Custom** tab to change the default properties.

To modify custom options for a change request

- 1 Click the **Custom** tab.
- 2 Double-click the name of the custom property. The **Edit Property** dialog box opens.
- 3 Select a new value for the property. For integer, text, and real fields, **Value** is a text box. For enumerated types and user IDs, it is a list box. For dates and times, **Value** has a **Date** check box and a **Time** check box, each of which is followed by a date or time in the format for your locale.
- 4 Repeat steps 2 and 3 for other custom properties.
- 5 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Descriptions](#)

[Specifying Change Request Solutions](#)

[Including Change Request Attachments](#)

[Adding Change Request Comments](#)

Related Reference

[Change Requests](#)

Resolve Open Change Requests

You resolve open change requests by following the steps below. Before you start working on a change request, be aware of any processes required by your team. For example:

- ◆ Your company might require that the change request **Status** be changed to **In Progress**.
- ◆ You might be required to link open change requests to the associated file or files that need to be changed. If this is the case, when you check in a file or group of files, you can indicate the change requests that are being fixed by the files. Doing this saves the time it would take to change the status of each change request.

To resolve a change request

- 1 Select the appropriate folder and click the **Change Requests** tab from the upper pane.
- 2 Optionally, expand or limit the number of change requests that are displayed by doing the following:
 - ◆ Select the **All Descendants** button on the toolbar.
 - ◆ Create a filter or query to limit the displayed change requests. For instance, you can find all change requests with **Open** status or simply sort the **Status** column in the upper pane to find all **Open** change requests.
- 3 Double-click the change request you want to resolve. The **Change Request <number, revision #>** dialog box opens.
- 4 Change the **Status** of the change request to one of the resolved statuses: **Fixed**, **Documented**, or **Cannot Reproduce**. Alternatively, you might use **Is Duplicate** or **As Designed**, if either of these is appropriate.

When you select a resolved status, StarTeam automatically makes the following changes to the change request:

- ◆ Places the name of the person who submitted the change request in the **Responsibility** field. The assumption is that the person who submitted the change request will want to know about, verify, or perhaps challenge the change in status.
- ◆ Changes the setting for the **Addressed in build** field to **Next Build** (if the status has changed to **Fixed** or **Documented**). When the next build label is created, **Next Build** changes to the name of the build label. The assumption is that the person who verifies that the change request has been implemented should test the correct build of the product.

- 5 If you choose **Fixed** or **Documented** as the new status, select the **Solution** tab and type the appropriate information in the **Work Around** and/or **Fix** text boxes.

Often a change request suggests one or more fixes for a problem, and none of these suggestions are implemented. To avoid confusion, the fix that is implemented must be described in precise detail. Testers and writers rely heavily on this information.

- 6 Click **OK** to close this dialog.

Tip: Although the application makes these automatic changes immediately, you can change the **Responsibility** or **Addressed in build** setting before you click **OK** (or **Apply**, if appropriate). In this way, you can bypass the automatic workflow and route the change request as your team requires.

Related Concepts

[Change Requests](#)

Related Procedures

[Verifying Resolved Change Requests](#)

[Closing Verified Change Requests](#)

Related Reference

[Change Requests](#)

Reviewing Linked Change Requests

If you are checking in a file that has one or more linked change requests, you should also review all change requests associated with the file.

To review a linked change request

- 1 Select the file in the upper pane that is linked to the change request.
- 2 Choose **File** ► **Check In**.
This opens the **Check In** dialog box.
- 3 Click **Advanced** to open the **Advanced Options** dialog box.
- 4 Click **Show Change Requests**.

The **Advanced Options** dialog box expands at the bottom and displays the list **Change Requests Linked In This View**.

Note: No change request appears in the list more than once, even if it is linked to several of the files you are checking in. When a change request is linked to more than one file, the list displays the name of only one of the files.

- 5 Optionally, double-click a change request to review or edit its properties.
- 6 Optionally, check **Marked Selected Change Requests As Fixed**.
If you check this option, StarTeam will mark the selected, but unresolved, change request **Fixed** as part of the check-in process.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Selecting Change Requests Using a Query

You can use a simple or complex query to limit the change requests displayed to those that fit specific criteria.

To select change requests using a query

- 1 Click **Change Request** tab in the upper pane.
- 2 Right-click a column header and choose **Queries** from the context menu.
- 3 Choose an existing query in the **Queries** dialog.
- 4 Click **Select** to apply it to the list of change requests.
- 5 Follow the steps in “Creating Queries” to define a new query.

Related Concepts

[Change Requests](#)
[Queries](#)

Related Procedures

[Customizing Change Request Reports](#)
[Creating Queries](#)
[Working with Change Requests](#)

Showing Fields in a Change Request

You can select which fields are displayed for a change request.

To show specific fields in a change request

- 1 Right-click the column headers and select **Show Fields** from the context menu.
This opens the **Show Fields** dialog box.
- 2 Make sure the **CR Number**, **Entered By**, and any other appropriate fields are displayed in the **Show these fields in this order** list.
- 3 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Sorting and Grouping Change Requests

You can sort change requests by the data in a particular column or group the change requests.

To sort or group change requests

- 1 Click on a column header on the Change Requests tab to sort the change requests by the data in that column.

Note: If you want to sort or group the change requests first by the data in one column and then by the data in another column, see the following steps. You can sort or group the change requests in up to four levels of groupings.

- 2 To sort or group the change requests in multiple levels, right-click the column headers and select **Sort and Group** from the context menu. The **Sort and Group** dialog box opens.
- 3 From the **First By** drop-down list box, select a column title.
- 4 Optionally, group the change requests by the data in this column, select the **Group By** check box.

If you select the **Group By** check box, the change requests are grouped together in nested lists and you must drill down to view the change requests in each group. If you do not select the **Group By** check box, the change requests are all displayed on the **Change Requests** tab, sorted by your choices in **Sort and Group** dialog box.

Note: By default, the column data is sorted or grouped based on the internal key or order. You can use the **Sort Options** button and choose to sort or group the data based on the text and optionally, case sensitivity.

- 5 Repeat steps three and four to define up to four levels of sort orders or groupings.
- 6 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Specifying Change Request Descriptions

You use the **Description** tab to specify detailed information about the change request including the steps to reproduce the problem.

To specify a description for a change request

- 1 Click the **Description** tab.
- 2 Type a detailed description of the change request in the **Description and steps to reproduce** text box.
Include the steps to reproduce the problem, or in the case of an enhancement request, a detailed description of the enhancement.
- 3 Optionally, type or browse for the path to a test for the change request in the **Test command** text box.
- 4 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Solutions](#)

[Modifying Custom Options for Change Requests](#)

[Including Change Request Attachments](#)

[Adding Change Request Comments](#)

Related Reference

[Change Requests](#)

Specifying Change Request Solutions

You use the **Solution** tab to specify a workaround for the problem and to document how this change request was resolved.

To specify a solution for a change request

- 1 Click the **Solution** tab.
- 2 Optionally, in the **Work around** text box, type the steps you can follow to work around the problem.
- 3 Optionally, in the **Fix** text box, type the solution to the problem. The **Fix** text box is usually completed by the user who fixes the code. In this box, the application can accept a maximum of 20K characters, but your database may accept fewer characters.
- 4 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Summary Information](#)

[Specifying Change Request Descriptions](#)

[Modifying Custom Options for Change Requests](#)

[Including Change Request Attachments](#)

[Adding Change Request Comments](#)

Related Reference

[Change Requests](#)

Specifying Change Request Summary Information

You use the **Synopsis** tab to define and modify the summary information about a change request. Summary information includes important criteria like status, severity, and who is currently responsible for this change request.

To specify the summary information for a change request

- 1 On the **Synopsis** tab, accept the default status **New** or select another status from the **Status** drop-down list box.
- 2 Indicate the severity of the change request by selecting **High**, **Medium**, or **Low** from the **Severity** drop-down list box.

Note: The team leader usually sets the criteria for high, medium and low status.

- 3 If the change request needs immediate attention, select **Yes** from the **Priority** list box.
- 4 To specify the type of change request, select **Defect** or **Suggestion** from the **Type** list box.
- 5 Select the platform to which the change request applies from the **Platform** drop-down list box.
- 6 Type a summary of the change request in the **Synopsis** text box. The application can accept a maximum of 20K characters in this text box, but your database may accept fewer characters.
- 7 Select the name of the team member responsible for correcting the change request from the **Responsibility** drop-down list box.
- 8 Click **Apply** to save your changes.

Related Concepts

[Change Requests](#)

Related Procedures

[Specifying Change Request Descriptions](#)

[Specifying Change Request Solutions](#)

[Modifying Custom Options for Change Requests](#)

[Including Change Request Attachments](#)

[Adding Change Request Comments](#)

Related Reference

[Change Requests](#)

Verifying Resolved Change Requests

You verify resolved change requests by following the steps below. If you determine that a change request is not really resolved, you can reopen it.

To mark a change request as verified or to reopen it

- 1 Select the appropriate folder and click the **Change Requests** tab from the upper pane.
- 2 Optionally, expand or limit the number of change requests that are displayed by doing the following:
 - ◆ Select the **All Descendants** button on the toolbar.
 - ◆ Create a filter or query to display the appropriate change requests or simply sort the **Status** column in the upper pane to group the change requests.
- 3 Double-click the change request you want to verify. The **Change Request <number, revision #>** dialog box opens.
- 4 Change the status to **Open** or **Verified**. StarTeam has the following verified statuses:
 - Verified As Designed**
 - Verified Cannot Reproduce**
 - Verified Documented**
 - Verified Fixed**
 - Verified Is Duplicate**
- 5 If you change the status to **Open**, type the word "Reopen" and the date in the **Synopsis** text box. Otherwise, the team member who resolved the change request may think that he or she forgot to mark it resolved and, without investigating further, mark it resolved a second time.

When you re-open a change request, StarTeam automatically does the following:

- ◆ Places the name of the person who resolved the change request in the **Responsibility** field. The assumption is that the person who resolved the change request the first time should be the person to continue working on it.
 - ◆ Blanks out the setting for the **Addressed in build** field. The assumption is that the change request has not been resolved and, therefore, has not been addressed in any build.
- 6 Do one of the following:
 - ◆ Click **Apply**, then click the **Next** or **Previous** button to verify another change request.
 - ◆ Click **OK** to close this dialog.

Related Concepts

[Change Requests](#)

Related Procedures

[Resolve Open Change Requests](#)

[Closing Verified Change Requests](#)

Related Reference

[Change Requests](#)

Working with Change Request Lists

Change requests are listed on the Change Request tab in the upper pane. You can move the columns around, sort the change requests by the data in a column, and change the display based on a filter or query. The sections below explain how to:

- ◆ Display the list of change requests for a folder.
- ◆ View unread change requests.
- ◆ Move change requests.
- ◆ Share change requests.

To display the list of change requests

- 1 Select a folder from the StarTeam folder hierarchy.
- 2 Click the **Change Requests** tab on the upper pane. A list of change requests for the current folder displays in the upper pane of the project view window. The **Change Request** menu item also becomes available on the menu bar.

All change requests shown in the upper pane:

- ◆ Are attached to the folder selected from the StarTeam folder hierarchy.
- ◆ Match the filter selected from the **Filter** drop-down list box.
- ◆ Match the depth specified by the **All Descendants** button.

To view unread change requests

- 1 Look for change requests in bold type. These are the change requests you are responsible for but have not yet reviewed.

Note: Change requests in regular type are those that you have read or those for which you are not responsible.

- 2 Click on the Responsibility column header, then scroll down to your name. All change requests in bold are unread.

To move change requests

- 1 Locate the change request you want to move. You can move a change request from one folder to another.
- 2 Click on the change request and drag it to a new folder.

To share change requests

- 1 Ctrl-click the change request and drag it to the folder that you want to share the change request.

Note: You can also share change requests with other views and projects as long as they use the same server configuration.

- 2 Click OK to confirm that you want to share the item.

Related Concepts

[Change Requests](#)

[User Roles and StarTeam Documentation](#)

Related Procedures

[Creating Change Requests](#)

[Creating Filters](#)

[Creating Queries](#)

[Working with Folders and Items](#)

Related Reference

[Change Requests](#)

Using Requirements

This section contains tasks related to using requirements.

In This Section

[Creating Requirements](#)

Describes how to create a requirement in StarTeam.

[Marking Item Threads Read or Unread](#)

Describes how to designate item threads as read or unread for requirement, task, and topic components.

Creating Requirements

Creating a hierarchy of requirements allows you to organize a project efficiently and work toward agreed-upon goals.

To create a requirement

- 1 Click the **Requirement** tab on the upper pane.
- 2 Do one of the following:
 - ◆ Choose **Requirement New** to create a new requirement that is not the child of an existing requirement.
 - ◆ Choose **Requirement New Child Requirement** to create a new requirement that is the child of an existing requirement.

The **New Requirement** dialog box appears.

- 3 Click the **Requirement** tab of the **New Requirement** dialog box and do the following:
 - ◆ Type a name for the requirement.
 - ◆ Select an owner (i.e., the person ultimately responsible for the fulfillment of the requirement) from the **Owner** drop-down list box.
 - ◆ Optionally, provide an external source or reference for the requirement in the **External reference** text box. If you publish requirements from CaliberRM to StarTeam, this field displays the CaliberRM identification for this requirement.
 - ◆ Type the initial description of this requirement in the **Description** text box. This description is usually revised over time to eliminate ambiguities.
- 4 Click the **Responsibility** tab, list the team members responsible for this requirement. If notification is enabled, these people will be notified about changes made to any field in the requirement.
 - ◆ Click **Add** to display the **Select Responsible Users** dialog box.
 - ◆ Double-click the name of each person to be added to the list. When you double-click the name, it moves from the **Users** list box to the **Responsible Users** list box.
 - ◆ Add the remaining responsible users to the **Responsible Users** list box and click **OK**.

Note: Ignore the **Ambiguity Review** tab for the time being. Later reviewers will use this tab to locate ambiguities in the initial description and revise that description.

- 5 Use the **Estimate** tab to indicate the best-case and worst-case times for fulfilling this requirement. The entries are usually in staff days.
 - ◆ Type the number of units (usually days) estimated for the fulfillment of this requirement in the **Expected effort** text box.
 - ◆ Type the number of units (usually days) estimated for the worst-case fulfillment of this requirement in the **High effort** text box.
 - ◆ Type the number of units (usually days) estimated for the best-case fulfillment of this requirement in the **Low effort** text box.
 - ◆ Add any appropriate notes in the **Notes** text box.
- 6 Use the **Custom** tab to provide values for any custom requirement properties that your team leader or company may have created. To change the default property:

- ◆ Double-click the name of the custom property. The **Edit Property** dialog box appears.
 - ◆ Select a new value for this property. For integer, text, and real fields, **Value** is a text box. For enumerated types and user IDs, it is a list box. For dates and times, **Value** has a date check box and a time check box, each of which is followed by a date or time in the format for your locale.
 - ◆ Repeat the above two steps for other custom properties.
- 7 Optionally, use the **Attachments** tab to add additional files to this requirement.
- ◆ Click **Add** if you want to attach a file. The **Open** dialog box appears.
 - ◆ In the **Open** dialog box, select the file to be attached.
 - ◆ Click **Open**.
 - ◆ Repeat the above three steps for additional attachments.
- 8 Use the **Comment** tab to explain why the requirement is being created or revised. Enter your reasons in the **Comment for new revision** text box.
- 9 Click **OK** to save the requirement.

Related Concepts

[Requirements](#)

Related Reference

[Requirement Properties](#)

Marking Item Threads Read or Unread

You can display requirement, task, and topic components in either a list display or hierarchical format. StarTeam provides a default view for requirement, task, and topic components in a hierarchical structure. When you select to view the hierarchical structure, each new item becomes the root of a tree. Its branches are child requirements, subtasks to the parent task, or responses to the topic. Children of children requirements, subtasks of subtasks, and responses to responses form additional branches. When an item thread is unread, the textual information about the item thread displays in bold text in upper pane. When an item thread is read, the item thread displays in normal text in the upper pane.

The main menu or context menu commands enable you to mark item threads as read or unread to better track these components.

To mark item threads read or unread

- 1 Select a topic thread in the upper pane.

Tip: To select multiple threads, display the information in list format by clicking the **List Display** button in the toolbar.

- 2 Do one of the following:

- ◆ Choose **Mark Thread as Read** to remove the bold format for the item thread.
- ◆ Choose **Mark Thread as Unread** to add bold format for the item thread.

Related Procedures

[Using Tasks](#)

[Using Requirements](#)

[Using Topics](#)

Using Tasks

This section contains procedures related to using StarTeam tasks.

In This Section

[Add Notes to Tasks](#)

Describes how to add notes to tasks.

[Adding Comments to Task Revisions](#)

Describes how to add revision comments to tasks.

[Assigning Task Resources](#)

Describes how to assign resources to StarTeam tasks and subtasks.

[Creating Tasks](#)

Describes how to create tasks and subtasks.

[Customizing Tasks](#)

Describes how to use customized fields to customize tasks.

[Marking Item Threads Read or Unread](#)

Describes how to designate item threads as read or unread for requirement, task, and topic components.

[Scoping Tasks](#)

Describes how to specify the amount of time it will take to complete a task

[Working with Attachments in Tasks](#)

Describes how to use attachments with tasks or subtasks.

[Working with Work Records in Tasks](#)

Describes how to add a work record to a task, and edit or delete previously entered work records.

Add Notes to Tasks

When you use the task component, you can enter additional information about the task or subtask on the **Notes** tab in the **Task Properties** dialog box.

To add notes to a task or subtask

- 1 Select a folder from the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following:
 - ◆ Double-click a task or subtask in the upper pane.
 - ◆ Right-click a task in the upper pane and choose **Properties**.
- 3 Click the **Notes** tab and type notes about the task in the **Notes** text box.
- 4 Click **OK** to save the notes.

Related Concepts

[Tasks](#)

[Links: Internal and External](#)

Related Procedures

[Creating Tasks](#)

[Assigning Task Resources](#)

[Scoping Tasks](#)

[Customizing Tasks](#)

[Working with Attachments in Tasks](#)

[Working with Work Records in Tasks](#)

[Adding Comments to Task Revisions](#)

[Viewing or Modifying Item Properties](#)

Adding Comments to Task Revisions

After you modify one or more properties in the **Task Properties** dialog box and click **OK**, the application creates a new revision of the task. You should add a revision comment or note explaining why you made the revision prior to clicking **OK** in the dialog box.

To provide a revision comment for a task

- 1 Double-click a task in the upper pane to open the **Task Properties** dialog box.
- 2 Make the desired changes to the task properties.
- 3 Click the **Comments** tab and type a revision comment.
- 4 Click **OK** to commit the changes and close the dialog box.

Related Concepts

[Tasks](#)

[Links: Internal and External](#)

Related Procedures

[Creating Tasks](#)

[Assigning Task Resources](#)

[Scoping Tasks](#)

[Customizing Tasks](#)

[Add Notes to Tasks](#)

[Working with Attachments in Tasks](#)

[Working with Work Records in Tasks](#)

[Viewing or Modifying Item Properties](#)

Assigning Task Resources

As you create a task or subtask, you can assign additional team members as resources to assist in the completion of the task. Use the **Resources** tab to review the list of team members available for this task assignment.

To assign team members to a task or subtask

- 1 Select a folder from the hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following to open the **Task Properties** dialog box:
 - ◆ Double-click a task or subtask in the **Task** pane.
 - ◆ Select a task or subtask in the **Task** pane and choose **Task ▸ Properties**.
 - ◆ Right-click a task or subtask in the **Task** pane and choose **Properties**.
- 3 Select the **Resources** tab and click **Add**.
The **Select Task Resources** dialog box opens.
- 4 Select the team members you want to assign to the task in the **Users** list, and click **Add**.
The selected names are moved from the **Users** list to the **Assigned Resources** list.
- 5 Click **OK** to return to the **Task Properties** dialog box.
- 6 Click **Apply** to apply the changes and keep the dialog box open, or click **OK** to close the dialog box.

To remove team members from a task or subtask

- 1 Select a folder from the hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following:
 - ◆ Double-click a task or subtask in the **Task** pane.
 - ◆ Select a task or subtask in the **Task** pane and choose **Task ▸ Properties**.
 - ◆ Right-click a task or subtask in the **Task** pane and choose **Properties**.
The **Task Properties** dialog box opens.
- 3 Click the **Resources** tab and select a resource name from the **Task Resource Assignments** list.
- 4 Click **Remove**.
- 5 Click **Apply** to apply the changes and keep the dialog box open, or click **OK** to close the dialog box.

Note: You can also remove a team member from the **Assigned Resources** list in the **Select Task Resources** dialog box by selecting a name in the **Assigned Resources** list and clicking **Remove** to return it to the **Users** list.

Related Concepts

[Tasks](#)

[Links: Internal and External](#)

Related Procedures

[Creating Tasks](#)

[Scoping Tasks](#)

[Customizing Tasks](#)

[Add Notes to Tasks](#)

[Working with Attachments in Tasks](#)

[Working with Work Records in Tasks](#)

[Adding Comments to Task Revisions](#)

[Viewing or Modifying Item Properties](#)

Creating Tasks

To create a task

1 Select a folder in the folder hierarchy and click the **Task** tab in the upper pane.

2 Do one of the following to open the **New Task** dialog box:

- ◆ Click the **New** button on the toolbar.
- ◆ Choose **Task ▸ New** on the menu.
- ◆ Right-click in the upper pane and choose **New**.

3 On the **Task** page of the **New Task** dialog box, type the **Name** of the task.

You can use up to 255 characters for the task name.

Note: Although the **Responsibility** field names the person primarily responsible for the completion of a task, additional people can be designated on the **Resources** tab.

4 Optionally, check **Milestone** to indicate that the task should be treated as a milestone task.

You can display the **Milestone** column in the task list and sort for the tasks that have been designated as milestones.

5 Select the current status of the task from the **Status** drop-down list.

The statuses are:

- ◆ **Pending:** This task is awaiting the completion or start of a predecessor task (Default).
- ◆ **Ready To Start:** This task may be started since all predecessor tasks have been completed.
- ◆ **In Progress:** Work has been entered for this task.
- ◆ **Finished:** This task is finished according to the team members working on the task.
- ◆ **Closed:** The task is completed to the satisfaction of the team leader or task reviewer.
- ◆ **Hold:** The task is completed to the satisfaction of the team leader or task reviewer.

6 Choose a **Priority** level from the drop-down list.

The priorities are identical to those used in Microsoft Project.

Note: Do not use the priority **Do Not Level**. This priority is a Microsoft Project-specific term.

7 Type the **Duration** which is the number of hours the task will take to complete.

Note: This field is disabled if the task contains subtasks since the duration of a task is dependent upon the duration of its subtasks.

8 Type the **Percent Complete** which is the percentage of work already completed for this task.

This field may range from 0-100. The default is 0 for new tasks.

9 Optionally, check **Needs Attention** to notify team leaders or task reviewers that this task requires attention.

Explain why this task needs attention in the text box. Team leaders can add the **Needs Attention** column to their task list and sort for items with this designation.

Click **OK** when finished.

This creates a new task which will serve as the root of a task tree in the **Task** pane.

To create a subtask

- 1 Select a folder in the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Select the task for which this will be a subtask and do one of the following to open the **New Task** dialog box:
 - ◆ Choose **Task** ► **New Subtask** on the menu.
 - ◆ Right-click the selected task in the upper pane and choose **New Subtask**.
- 3 Complete the **Task** page of the **New Task** dialog box as you did for the task.
- 4 Click **OK** when finished.

This creates a new subtask for the selected task. The name of the task for which this is a subtask is displayed above the subtask name field.

Note: Additional tabs in the **New Task** dialog box provide additional tracking options and features.

Related Concepts

[Tasks](#)

[Links: Internal and External](#)

Related Procedures

[Assigning Task Resources](#)

[Scoping Tasks](#)

[Customizing Tasks](#)

[Add Notes to Tasks](#)

[Working with Attachments in Tasks](#)

[Working with Work Records in Tasks](#)

[Adding Comments to Task Revisions](#)

[Viewing or Modifying Item Properties](#)

Customizing Tasks

If your administrator has created custom fields for the task component, you may be required to complete these fields when entering or modifying tasks or subtasks. The availability of custom fields depends upon your application license.

To customize tasks

- 1 Select a folder in the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following to open the **Task Properties** dialog box:
 - ◆ Double-click a task or subtask in the **Task** pane.
 - ◆ Select a task or subtask in the **Task** pane and choose **Task ▸ Properties**.
 - ◆ Right-click a task or subtask in the **Task** pane and choose **Properties**.
- 3 Double-click a custom property on the **Custom** tab to open the **Edit Property** dialog box.
- 4 Type or select a new value for the property:
 - ◆ For integer, text, and real fields, **Value** is a text box.
 - ◆ For enumerated types and user IDs, **Value** is a list box.
 - ◆ For dates and times, **Value** has a **Date** check box and a **Time** checkbox, each of which is followed by a date or time in the format for your locale.
- 5 Click **OK**.

Related Concepts

[Tasks](#)

[Links: Internal and External](#)

Related Procedures

[Creating Tasks](#)

[Assigning Task Resources](#)

[Scoping Tasks](#)

[Add Notes to Tasks](#)

[Working with Attachments in Tasks](#)

[Working with Work Records in Tasks](#)

[Adding Comments to Task Revisions](#)

[Viewing or Modifying Item Properties](#)

Marking Item Threads Read or Unread

You can display requirement, task, and topic components in either a list display or hierarchical format. StarTeam provides a default view for requirement, task, and topic components in a hierarchical structure. When you select to view the hierarchical structure, each new item becomes the root of a tree. Its branches are child requirements, subtasks to the parent task, or responses to the topic. Children of children requirements, subtasks of subtasks, and responses to responses form additional branches. When an item thread is unread, the textual information about the item thread displays in bold text in upper pane. When an item thread is read, the item thread displays in normal text in the upper pane.

The main menu or context menu commands enable you to mark item threads as read or unread to better track these components.

To mark item threads read or unread

- 1 Select a topic thread in the upper pane.

Tip: To select multiple threads, display the information in list format by clicking the **List Display** button in the toolbar.

- 2 Do one of the following:

- ◆ Choose **Mark Thread as Read** to remove the bold format for the item thread.
- ◆ Choose **Mark Thread as Unread** to add bold format for the item thread.

Related Procedures

[Using Tasks](#)

[Using Requirements](#)

[Using Topics](#)

Scoping Tasks

The task component includes a **Time** tab on which you can record the amount of time needed to complete a task or subtask.

To enter the estimated time to complete a task

- 1 Select a folder in the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following to open the **Task Properties** dialog box:
 - ◆ Double-click a task or subtask in the **Task** pane.
 - ◆ Select a task or subtask in the **Task** pane and choose **Task ▸ Properties**.
 - ◆ Right-click a task or subtask in the **Task** pane and choose **Properties**.
- 3 Click the **Time** tab.
- 4 Use the **Start** and **Finish** buttons to select a start and finish date.
- 5 Type the estimated hours required to complete the task in the **Work** field.

The rest of the **Time** pane is disabled, however the values for **Actual** and **Variance** automatically calculate when the **Work** value changes.
- 6 Click **OK** when finished.

Related Concepts

[Tasks](#)

[Links: Internal and External](#)

Related Procedures

[Creating Tasks](#)

[Assigning Task Resources](#)

[Customizing Tasks](#)

[Add Notes to Tasks](#)

[Working with Attachments in Tasks](#)

[Working with Work Records in Tasks](#)

[Adding Comments to Task Revisions](#)

[Viewing or Modifying Item Properties](#)

Working with Attachments in Tasks

You can include screen shots and documentation with a task or subtask on the **Attachments** tab in the **Task Properties** dialog box. Attachments can also be renamed or removed from this tab.

To add an attachment to a task

- 1 Select a folder from the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following:
 - ◆ Double-click a task or subtask in the upper pane.
 - ◆ Right-click a task in the upper pane and choose **Properties**.
- 3 Click **Add** on the **Attachments** page.
This displays the **Open** file browser dialog box.
- 4 Select the file you want to attach to the task and click **Open**.
The selected item appears in the **Attachments** list in the on the **Attachments** page of the **Task Properties** dialog box.
- 5 Click **OK** to close the **Task Properties** dialog box.

To rename an attachment

- 1 Select a folder from the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following:
 - ◆ Double-click a task or subtask in the upper pane.
 - ◆ Right-click a task in the upper pane and choose **Properties**.
- 3 Click the **Attachments** page.
- 4 Select a file in the **Attachments** list, and click **Save As**.
- 5 In the **Save As** dialog box, type a new name for the file.
- 6 Click **Save**.
- 7 Click **OK** to close the **Task Properties** dialog box.

To remove an attachment

- 1 Select a folder from the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following:
 - ◆ Double-click a task or subtask in the upper pane.
 - ◆ Right-click a task in the upper pane and choose **Properties**.
- 3 Select the file you want to remove on the **Attachments** page and click **Remove**.
The attachment disappears from the list.

- 4 Click **OK** to close the **Task Properties** dialog box.

Related Concepts

[Tasks](#)

[Links: Internal and External](#)

Related Procedures

[Creating Tasks](#)

[Assigning Task Resources](#)

[Scoping Tasks](#)

[Customizing Tasks](#)

[Add Notes to Tasks](#)

[Working with Work Records in Tasks](#)

[Adding Comments to Task Revisions](#)

[Viewing or Modifying Item Properties](#)

Working with Work Records in Tasks

After working on a task or subtask, you should add a work record to indicate what was done and the time spent. For example, if you work on a task for one hour on Day 1 and for three hours on Day 2, you would enter two work records, one for each day. You can edit or delete previously entered work records

Warning: After you have added a work record to a task, you cannot create subtasks for that task.

To add a work record to a task

- 1 Select a folder from the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following:
 - ◆ Double-click a task or subtask in the upper pane to display the **Task Properties** dialog box. Click **Add** on the **Work** page.
 - ◆ Select a task and choose **Task** ► **Add Work**.

The **Work Record** dialog box opens and displays your **User Name** in the drop-down list at the top.

- 3 Click the **Date** button and select a date for the work record.
- 4 Type the number of hours worked in the **Work** text box.
- 5 Type the number of hours it will take to complete the task in the **Remaining Work** text box.
- 6 Type comments about the progress that has been made in the **Comments** text box.
- 7 Click **OK** to save the work record.

To edit a work record on a task

- 1 Select a folder from the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following:
 - ◆ Double-click a task or subtask in the upper pane.
 - ◆ Right-click a task in the upper pane and choose **Properties**.

This opens the **Task Properties** dialog box.

- 3 On the **Work** page, select a record from the **Work Records** list and click **Edit**.
This opens the Work Record dialog box.
- 4 In the **Work Records** dialog box, make any changes to the work record.
- 5 Click **OK** to close the **Task Properties** dialog box.

To delete a work record from a task

- 1 Select a folder from the folder hierarchy and click the **Task** tab in the upper pane.
- 2 Do one of the following:
 - ◆ Double-click a task or subtask in the upper pane.
 - ◆ Right-click a task in the upper pane and choose **Properties**.

This opens the **Task Properties** dialog box.

- 3 On the **Work** page, select a record from the **Work Records** list and click **Delete**.
The message, "Delete Work Record?" appears. Click **Yes** to confirm the deletion.
- 4 Click **OK** to close the **Task Properties** dialog box.

Related Concepts

[Tasks](#)

[Links: Internal and External](#)

Related Procedures

[Creating Tasks](#)

[Assigning Task Resources](#)

[Scoping Tasks](#)

[Customizing Tasks](#)

[Add Notes to Tasks](#)

[Working with Attachments in Tasks](#)

[Adding Comments to Task Revisions](#)

[Viewing or Modifying Item Properties](#)

Using Topics

This section contains tasks related to using topics.

In This Section

[Creating Topics](#)

Describes how to create a topic.

[Marking Item Threads Read or Unread](#)

Describes how to designate item threads as read or unread for requirement, task, and topic components.

[Responding To Topics](#)

Describes how to respond to a topic.

Creating Topics

To start a threaded conversation, you must first create a topic.

To create a topic

- 1 Select a folder from the project tree.
- 2 Select the **Topic** tab.
- 3 Choose **New** from the **Topic** menu or context menu.
The **New Topic** dialog box opens.
- 4 Click the **Topic** tab and type the title of your topic in the **Title** text box.
- 5 Type the content for this topic in the **Content** text box.
- 6 Optionally, do any of the following:
 - ◆ Use the **Options** tab if you want to send the topic to specific team members, assign a priority to the topic, or indicate a status for the topic.
 - ◆ If your administrator created additional topic properties, you can access them on the **Custom** tab.
 - ◆ Use the **Attachments** tab if you want to attach a file or graphic with your topic.

To perform any of these actions, see the instructions below.

- 7 Optionally, select the **Comment** tab to add additional notes or a comment in the **Comment for new revision** text box.
- 8 When the information is complete, click **OK**. This action enters the new topic in the upper pane of the **Topics** component.

If the tree format is selected, the topic title, your user name, and the time stamp display. If the list format is selected, the list displays the same information, but includes one additional column, **Description**, which shows the first few words in the topic text.

To send a topic to additional team members, set the priority, or set the status

- 1 Select the **Options** tab.
- 2 Click **Add**. The **Select Topic Recipients** dialog box opens.
- 3 Select the team members from the list, then click **Add**.

Note: If you change a topic to delete its recipients, you cannot delete yourself as a recipient unless you delete all of the recipients. When there are recipients, StarTeam does not allow you to remove yourself from the notification about it.

- 4 To assign a priority to the topic, select **Low**, **Normal**, or **High** from the **Priority** drop-down list box.
- 5 To specify a topic status, select either **Active** or **Inactive** from the **Status** drop-down list box. The default status is **Active**.
- 6 Click **OK** to return to the **New Topic** dialog box.

To set the values for custom properties

- 1 Select the **Custom** tab.
- 2 Double-click the name of a custom property. The **Edit Property** dialog box opens.

- 3 Select a new value for this property. For integer, text, and real fields, **Value** is a text box. For enumerated types and user IDs, it is a list box. For dates and times, **Value** has a date check box and a time check box, each of which is followed by a date or time in the format for your locale.
- 4 Repeat the previous two steps for other custom properties.

To attach a file or graphic with your topic

- 1 Select the **Attachments** tab.
- 2 Click **Add**. The **Open** dialog box displays.
- 3 Type the attachment name in the **File name** text box.
- 4 Click **Open**.
- 5 Repeat the previous steps for additional attachments.

Related Concepts

[Topics](#)

Related Procedures

[Using Topics](#)

[Responding To Topics](#)

Related Reference

[Topic Options \(Personal Options Dialog Box\)](#)

Marking Item Threads Read or Unread

You can display requirement, task, and topic components in either a list display or hierarchical format. StarTeam provides a default view for requirement, task, and topic components in a hierarchical structure. When you select to view the hierarchical structure, each new item becomes the root of a tree. Its branches are child requirements, subtasks to the parent task, or responses to the topic. Children of children requirements, subtasks of subtasks, and responses to responses form additional branches. When an item thread is unread, the textual information about the item thread displays in bold text in upper pane. When an item thread is read, the item thread displays in normal text in the upper pane.

The main menu or context menu commands enable you to mark item threads as read or unread to better track these components.

To mark item threads read or unread

- 1 Select a topic thread in the upper pane.

Tip: To select multiple threads, display the information in list format by clicking the **List Display** button in the toolbar.

- 2 Do one of the following:

- ◆ Choose **Mark Thread as Read** to remove the bold format for the item thread.
- ◆ Choose **Mark Thread as Unread** to add bold format for the item thread.

Related Procedures

[Using Tasks](#)

[Using Requirements](#)

[Using Topics](#)

Responding To Topics

After someone starts a topic, you can reply to the topic or to one or more of its responses.

To respond to a topic or response

- 1 On the **Topic** tab in the upper pane, select the item to which you are responding.
- 2 Choose **Respond** on the **Topic** menu or context menu. The **New Topic** dialog box opens.
- 3 On the **Topic** tab in the **New Topic** dialog box, type the title of your response in the **Title** text box.
- 4 Type your remarks in the **Content** text box.
- 5 Optionally, do any of the following:
 - ◆ Use the **Options** tab if you want to send the topic to specific team members, assign a priority to the topic, or indicate a status for the topic.
 - ◆ If your administrator created additional topic properties, you can access them on the **Custom** tab.
 - ◆ Use the **Attachments** tab if you want to attach a file or graphic with your topic.

To perform any of these actions, see the instructions below.

- 6 Optionally, select the **Comment** tab to add additional notes or a comment in the **Comment for new revision** text box.
- 7 When the information is complete, click **OK**. This action enters the new topic in the upper pane of the **Topics** component.

If the tree format is selected, the response appears in relation to other parts of the threaded conversation. The response title, your user name, and the time stamp also display. If the list format is selected, the list displays the same information, but includes one additional column, **Description**, which shows the first few words in the response text.

To send a topic to additional team members, set the priority, or set the status

- 1 Select the **Options** tab.
- 2 Click **Add**. The **Select Topic Recipients** dialog box opens.
- 3 Select the team members from the list, then click **Add**.
- 4 To assign a priority to the topic, select **Low**, **Normal**, or **High** from the **Priority** drop-down list box.
- 5 To specify a topic status, select either **Active** or **Inactive** from the **Status** drop-down list box. The default status is **Active**.
- 6 Click **OK** to return to the **New Topic** dialog box.

To set the values for custom properties

- 1 Select the **Custom** tab.
- 2 Double-click the name of a custom property. The **Edit Property** dialog box opens.
- 3 Select a new value for this property. For integer, text, and real fields, **Value** is a text box. For enumerated types and user IDs, it is a list box. For dates and times, **Value** has a date check box and a time check box, each of which is followed by a date or time in the format for your locale.
- 4 Repeat the previous two steps for other custom properties.

To attach a file or graphic with your topic

- 1 Select the **Attachments** tab.
- 2 Click **Add**. The **Open** dialog box displays.
- 3 Type the attachment name in the **File name** text box.
- 4 Click **Open**.
- 5 Repeat the previous steps for additional attachments.

Related Concepts

[Topics](#)

Related Procedures

[Using Topics](#)

[Creating Topics](#)

Related Reference

[Topic Options \(Personal Options Dialog Box\)](#)

Viewing the Audit Log

This section contains tasks related to viewing the Audit Log.

In This Section

[Filtering Audit Log Entries](#)

Describes how to filter audit log entries.

[Searching for Log Entries](#)

Describes how to search for information in the audit logs.

[Sending Log Entries Through E-mail](#)

Describes how to e-mail an item from an audit log.

[Sorting Audit Log Entries](#)

Describes how to sort entries in the audit log.

Filtering Audit Log Entries

When you click the **Audit** tab, it displays audit log entries for the selected view in the upper pane of the project view window. The **Audit** menu item also becomes available on the menu bar. The list of audit records depends on your selection from the folder hierarchy in the left pane and whether the **All Descendants** button is selected from the toolbar or **Audit** menu.

If you select this tab and the upper pane is empty, your administrator has disabled the audit log.

Filtering allows you to specify what fields are displayed in the audit entries that appear in the upper pane and how those fields are grouped and sorted.

To filter audit log entries

- 1 Click on the **Filter** drop-down list box on the tool bar.
- 2 Select one of the following default filters:
 - ◆ <By Class and Event> — Displays audit entries sorted by their value in the Class Name 1 field (type of item) and Event (type of action) field.
 - ◆ By Transaction and Event — Groups audit log entries by descending **Transaction ID** and then by **Event** type. This filter provides a reverse-chronological list of updates in the view by transaction.
 - ◆ Events — Groups audit log entries by **Event** type, then by **Target 1 Class ID**, and then by **Created Time**.
 - ◆ Show All — displays all entries. This is the default option.

Note: You can customize the default filters or create new ones. See **Creating Filters** under **Related Procedures** below. You can also limit the number of audit log entries displayed by creating a query that selects audit log entries by specific property values. See **Queries** under **Related Concepts** below.

Related Procedures

[Creating Filters](#)

[Searching for Log Entries](#)

[Sending Log Entries Through E-mail](#)

[Sorting Audit Log Entries](#)

Searching for Log Entries

When you click on the Audit tab, it displays audit log entries for the selected view in the upper pane of the project view window. The **Audit** menu item also becomes available on the menu bar. If you select this tab and the upper pane is empty, your administrator has disabled the audit log.

All entries shown in the Audit list:

- ◆ Are associated with the folder selected from the StarTeam folder hierarchy
- ◆ Match the filter selected from the **Filter** drop-down list box
- ◆ Match the depth specified by the **All Descendents** button

To search for audit log entries do one of the following

- 1 Click on **Audit** ► **Find**
- 2 Click on **Audit** ► **Find Next**
- 3 Click on **Audit** ► **Find Previous**

Related Procedures

[Filtering Audit Log Entries](#)

[Sending Log Entries Through E-mail](#)

[Sorting Audit Log Entries](#)

Sending Log Entries Through E-mail

You can send an Audit list item as an e-mail, although the attachments in the item will not be sent.

To send an Audit list entry through e-mail

- 1 Select the item you want to send by clicking on it.
- 2 Click on **Audit** ► **Send To**. This will open the **Send To** dialog box.
This will send the selected item in the Audit list (except for the attached files) as an e-mail.

Related Procedures

[Searching for Log Entries](#)
[Filtering Audit Log Entries](#)
[Sorting Audit Log Entries](#)

Sorting Audit Log Entries

The sort usually takes place in descending or ascending numeric or alphanumeric order depending on the data.

To sort audit log entries

- 1 Click on an audit column header to perform a sort based on the value in that column.
- 2 To change the sort order from ascending to descending or vice versa, click the header a second time.

Related Procedures

[Searching for Log Entries](#)

[Filtering Audit Log Entries](#)

[Sending Log Entries Through E-mail](#)

Managing Labels and Promotion States

This section contains tasks related to managing labels and promotion states.

In This Section

[Attaching Labels to Folders](#)

Describes how to label folders.

[Attaching Labels to Items](#)

Describes creating new revision label for an item, reviewing all labels attached to item revisions, and moving a revision label from one item revision to another.

[Configuring or Viewing Label Properties](#)

Describes how to access label properties so you can view or modify them.

[Configuring Promotion States](#)

Describes how to create and configure a promotion state, and how to assign to it access rights.

[Copying Revision Labels](#)

Describes how to copy an existing revision label.

[Copying View Labels](#)

Describes how to create a view label based on an existing view label.

[Creating Revision Labels](#)

Describes how to create a new revision label.

[Creating View Labels](#)

Describes how to create a view label.

[Deleting Labels](#)

Describes how to completely remove a view or revision label.

[Demoting View Labels](#)

Describes how to demote a view label.

[Detaching a Label from a Rolled-back View](#)

Describes how to detach a label from a view that has been rolled-back to a previous revision.

[Detaching a Label from a Specific Revision](#)

Describes how to detach a label from a specific revision.

[Detaching Labels from Folders](#)

Describes how to detach view or revision labels from folders and their contents

[Detaching Labels from Items](#)

Describes how to detach items or folders from a view or revision label.

[Freezing or Unfreezing Labels](#)

Describes how to freeze or unfreeze labels.

[Promoting View Labels](#)

Describes how to promote a view label from one promotion state to the next.

[Reviewing and Moving Labels](#)

Describes how to review labels, and how to move labels from one item or folder revision to another.

[Sorting Labels Alphabetically](#)

Describes how to sort labels alphabetically

Attaching Labels to Folders

Labeling folders is slightly different from labeling items. When you attach a revision label to a folder, you can also attach it to the items that the folder contains and to everything in the subtree for which the folder is the root (its child folders and their contents).

If you detach a revision label from a folder, you can also detach the label from the items associated with the folder and, optionally, from the child folders and their items. If you detach a view label, the label is automatically detached from the items that the folder contains, from the child folders, and from their contents.

Note: To determine whether a label is a revision label or a view label, double-click the label (or select the label, and then click Properties). A revision label has a name and a description. A view label has a name, description, and a configuration time.

To create a new revision label and attach it to a folder and its contents

- 1 Select a folder in the folder hierarchy tree or on the **Folder** tab in the upper pane.
- 2 Right-click the selected folder and choose **Labels** to open the **Labels** dialog box.
- 3 Select the revision that will receive the new label.
- 4 Click **New**.
The **Attach a New Revision Label** dialog box opens.
- 5 Type a name and description for the label in the appropriate text boxes (up to 254 characters.)
- 6 Optionally, check **Frozen** (that is, cannot be changed) to ensure that only the selected revision can have this label.
- 7 Do one of the following:
 - ◆ Select **Folder Only** to attach a label to only the selected folder.
 - ◆ Select **Folder and Items Contained in Folder** to attach a label to the folder and its items.
 - ◆ Select **Everything in Subtree Rooted at Folder** to attach a label to the folder, its items, and its child folders and their items.

Note: Because attaching a label to a folder also allows it to be attached to the folder contents, children, and so on, the label is always attached to the current configuration of each folder and item. You cannot label a prior revision of a folder.

To attach an existing view or revision label to a folder and its contents

- 1 Select a folder in the folder hierarchy tree or on the **Folder** tab.
- 2 Right-click the selected folder and choose **Labels** to open the **Labels** dialog box.
- 3 Click **Attach**. The **Attach a Label** dialog box lists all the existing labels and identifies them as view or revision labels. By default, both the **View Labels** and **Revision Labels** check boxes are selected.
To display only view labels or revision labels, uncheck the appropriate check box.
- 4 Select a label.
- 5 Do one of the following:
 - ◆ Select **Folder Only** to attach a label to only the selected folder.
 - ◆ Select **Folder and Items Contained in Folder** to attach a label to the folder and its items.

- ◆ Select **Everything in Subtree Rooted at Folder** to attach a label to the folder, its items, and its child folders and their items.

Note: Attaching a label to a folder always attaches it to the current configuration of each folder and item. It is not possible to label a past revision of a folder, although you can do so for items.

6 Click **OK**.

To review the labels attached to a folder's revisions

- 1 Select a folder in the folder hierarchy tree or the **Folder** tab.
- 2 Right-click the selected folder and choose **Labels** to open the **Labels** dialog box.
The **Labels** dialog box lists all labels currently attached to this folder on a revision-by-revision basis.

To move a revision label from one folder revision to another

- 1 Select a folder in the folder hierarchy tree or on the **Folder** tab.
- 2 Right-click the selected folder and choose **Labels**.
- 3 Drag a revision label from one folder revision node to another in the **Labels** dialog box.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)
[Creating View Labels](#)
[Creating Revision Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Deleting Labels](#)
[Detaching Labels from Items](#)

Attaching Labels to Items

If you are dealing with an item or set of items that you want to group together, you can create a new revision label, attach an existing label to the item or an item revision, review all labels, or move a revision label.

To create a new revision label for selected items

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select one or more items in the upper pane.
- 3 Right-click the selected item(s) and choose **Labels** ► **New** to open the **Attach a New Revision** dialog box.
- 4 Type a name and description for the label in the appropriate text boxes. The maximum label name length is 64 characters and the description length is 254 characters.
- 5 Optionally, check **Frozen** to ensure that only the selected item revision can have this label.
- 6 Indicate what item revision is to receive this label by selecting a configuration option. The choices are:
 - ◆ **Current Configuration** to attach the label to the tip revision.
 - ◆ **Labeled Configuration** to attach the label to the revision with a specified label. The labels are in reverse chronological order based on the time at which they were created.
 - ◆ **Promotion State Configuration** to attach the label to the revision currently in a specified promotion state. (Actually, the label is attached to the revision that has the promotion state's current view label.)
 - ◆ **Configuration As Of** to attach the label to the revision that was the tip revision at a specified date and time.
- 7 Click OK.

To attach an existing view or revision label to selected items

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select one or more items in the upper pane.
- 3 Right-click the selected item(s) and choose **Labels** ► **Attach** to open the **Attach a Label** dialog box. This dialog box lists all existing labels and identifies them as view or revision labels. By default, both the **View Labels** and **Revision Labels** options are checked.
- 4 Uncheck **View Labels** or **Revision Labels** to limit the list to one specific type of label.
- 5 Select a label from the list.
- 6 Optionally, change the selection for what item revision is to receive this label by selecting a configuration option. The choices are:
 - ◆ **Current Configuration** to attach the label to the tip revision.
 - ◆ **Labeled Configuration** to attach the label to the revision with a specified label. The labels are in reverse chronological order based on the time at which they were created.
 - ◆ **Promotion State Configuration** to attach the label to the revision currently in a specified promotion state. (Actually, the label is attached to the revision that has the promotion state's current view label.)
 - ◆ **Configuration As Of** to attach the label to the revision that was the tip revision at a specified date and time.

- 7 Click **OK**.

To attach an existing view or revision label to a specific item revision

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select the item in the upper pane, then click the **Label** tab in the lower pane.
- 3 Right-click an item revision in the **Label** pane and choose **Attach** to open the **Attach a Label** dialog box. This dialog box lists all existing labels and identifies them as view or revision labels. By default, both the **View Labels** and **Revision Labels** options are checked.
- 4 Uncheck **View Labels** or **Revision Labels** to limit the list to one specific type of label.
- 5 Select a label from the list.
- 6 Optionally, change the selection for what item revision is to receive this label by selecting a configuration option. The choices are:
 - ◆ **Current Configuration** to attach the label to the tip revision.
 - ◆ **Labeled Configuration** to attach the label to the revision with a specified label. The labels are in reverse chronological order based on the time at which they were created.
 - ◆ **Promotion State Configuration** to attach the label to the revision currently in a specified promotion state. (Actually, the label is attached to the revision that has the promotion state's current view label.)
 - ◆ **Configuration As Of** to attach the label to the revision that was the tip revision at a specified date and time.

- 7 Click **OK**.

To review all labels attached to item revisions

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select the item in the upper pane, then click the **Label** tab in the lower pane. This action displays the **Label** pane which shows all revisions for the item.
- 3 Right-click a revision in the **Label** pane to display all of its labels.

To move a revision label from one item revision to another

- 1 Select a folder from the folder hierarchy on the left and click the component tab containing the items you want to label.
- 2 Select the item in the upper pane, then click the **Label** tab in the lower pane. This action displays the **Label** pane which shows all revisions for the item.
- 3 Right-click a revision in the **Label** pane to display all of its labels.
- 4 Select a specific label and drag it from one revision to the another.

Note: A Label can be attached to only one revision of an item.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Folders](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Copying Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Configuring or Viewing Label Properties

View label properties include a name, description, frozen/unfrozen status, configuration, and build label status. Revision label properties include a name, description, and frozen/unfrozen status.

To display view or revision label properties for editing

- 1 Select the appropriate folder in the folder hierarchy tree.
- 2 Choose **View** ► **Labels**.
This displays the **Labels** dialog box.
- 3 Select a label on the **View** or **Revision** tabs and click **Properties**.
The **Edit Label** dialog box opens enabling you to modify the label name or description, and freeze or unfreeze the label.

To display folder label properties

- 1 Select the appropriate folder in the folder hierarchy tree.
- 2 Right-click the selected folder and choose **Labels**.
This displays the **Labels** dialog box showing all the labels attached to the folder.
- 3 Select a label from the list and click **Properties**.
A read-only **View Properties** dialog box opens enabling you to view the properties for the selected label.

To display label properties from the Label pane

- 1 Select an item in the upper pane.
- 2 Click the **Label** tab in the lower pane.
- 3 Right-click a label in the **Label** pane and choose **Properties**.
A read-only **View Properties** dialog box opens enabling you to view the properties for the selected label.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Copying Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Configuring Promotion States

When creating promotion states, many administrators assign **<current>** to the initial promotion state instead of a view label, because that state always uses the tip revisions. They also often assign **<current>** to later promotion states for which no view labels currently exist. These states may receive a view label later, when the files associated with a view label meet the criteria required by the state. Alternatively, a view label may be promoted from the preceding state to the label-less state.

You can create promotion states only if you have the required access rights, which are found at the project or view level.

To create a new promotion state

- 1 Choose **View** ► **Promotion**.

The **Promotion** dialog box displays the states currently created for this view. The final state appears at the top of the list.

- 2 Click **Add** to open the **Promotion State** dialog box.
- 3 Type the **Name** and **Description** of the promotion state.
- 4 Assign a view label to this state by selecting it from the **View Label** drop-down list.

The labels are listed in reverse chronological order, based on the time at which they were created. You can change the label when appropriate by using this dialog box or assign it to the next state by promoting it.

- 5 Click **OK** to close the **Promotion State** dialog box, and **OK** again to close the **Promotion** dialog box.

To edit or delete a promotion state

- 1 Choose **View** ► **Promotion**.

The **Promotion** dialog box displays the states currently created for this view. The final state appears at the top of the list.

- 2 Click **Edit** to open the **Promotion State** dialog box.
- 3 Modify the **Name**, **Description**, or **View Label**.
- 4 Click **OK** to close the **Promotion State** dialog box, and **OK** again to close the **Promotion** dialog box.

To move the promotion state up or down in the list

- 1 Open the **Promotion** dialog box.
- 2 Select a promotion state from the list and click **Move Up** or **Move Down**.

To modify access rights to the promotion state

- 1 Open the **Promotion** dialog box.
- 2 Select a promotion state from the list and click **Access Rights**.
This opens the **Promotion State Access Rights** dialog box.
- 3 Do one of the following:

- ◆ Click **Add** to specify a group or specific user to grant or deny specific access privileges. Check the specific access rights and select **Grant** or **Deny**.

- ◆ Select an existing user or group and change which access privileges they have.

Related Concepts

[Promotion States](#)

[Proper Use of Views](#)

Related Procedures

[Managing Labels and Promotion States](#)

[Promoting View Labels](#)

[Assigning Access Rights to Projects](#)

Copying Revision Labels

Occasionally, you may want to copy a revision label. For example, if you move or share an item from one view (source view) to another (target view), labels from the source view do not become part of target view. However, by copying the revision labels after the move or share, you can selectively maintain revision labels on the moved or shared items.

Copying a revision label immediately attaches it to the same revisions of the same items as the original revision label. If the two revision labels are in the same view, each label will be attached to the same number of items. However, if the two revision labels are in different views, the new label becomes attached to the same revisions of the same items only if the items and their revisions exist in the new label's view at the time of the copy operation.

Although you can copy revision labels in a variety of ways, the following procedure allows you to copy a revision label whether it is in the current view or in another accessible view. It assumes that you are dealing with files, but can be adapted for other types of items.

To create a revision label based on an existing label (with a few additions or exceptions)

- 1 Choose **View** ► **Labels** and select the **Revision** tab in the **Labels** dialog box.
- 2 Click **New** to open the **Revision Label** dialog box.
- 3 Type a name and description for the revision label.
- 4 Check **Copy From Another Revision Label**.
- 5 Click **Select** to open the **Copy a Revision** dialog box.
- 6 Select a project from **Project** from the list, a view in the **View** tree, and a revision label in the **Labels** list.
- 7 Click **OK** to close the **Copy a Revision Label** dialog box, and click **OK** to close the **Revision Label** dialog box. Click **Close** on the **Labels** dialog box.

The new revision label is attached to the same revisions as the existing label.

- 8 Do one of the following:
 - ◆ Check in the changed file or files using the new revision label.
 - ◆ Check in the changed file and attach the new revision label manually to the changed file revisions. To do this, select the checked in file and click the **Label** tab in the lower pane. Drag the new revision label to the correct (probably tip) revision in the **Label** pane. Repeat for any other changed files.

Note: If you have added a new file, use **File** ► **Labels** ► **Attach** to attach the label.

Related Concepts

[Labels](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Attaching Labels to Folders](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Copying View Labels

Occasionally, you may want to create a view label and attach it to the same item revisions as an existing view label, with a few additions or exceptions. The steps in this procedure explain how to create a view label based on an existing view label. For example, suppose builds are done only after a view has been rolled back to a label and that the build is given the same name as the label. If, in the last build, only one Help file was missing, you would probably change the existing label to include that one file and rebuild. However, if the previous build was already made available to users participating in a field test, using the same label could cause confusion. It would be better to create a new view label as a copy of the older label and then add the missing file to the new label.

Note: You cannot copy a view label unless it already exists in the view in which you are performing this operation. The view configuration must also be current.

To create a new view label and attach it to the same item revisions as the existing view label

- 1 Choose **View** ► **Labels** to open the **Labels** dialog box.
- 2 Click **New** to open the **View Label** dialog box.
- 3 Type a name and description for the label.
- 4 Select the **Labeled Configuration** option to attach the label to item revisions that have an existing label.
- 5 Optionally, uncheck **Use As Build Label** if you do not want this label to be a build label.

Note: By default all view labels are designated as build labels.

- 6 Click **OK**, then click **Close** to exit the **Labels** dialog box.
The new view label is now attached to the same revisions as the existing label.
- 7 Select the items in the upper pane for which the new label must differ.

Tip: You can also select all items with a specific label. Right-click in the upper pane, choose **Select** ► **By Label**. When you select the label, all the items attached to that label are automatically selected.

- 8 Detach the new label from items that you do not want to include.
- 9 Attach the new label to items formerly not included, and/or attach the new label to different revisions of items to which it is already attached.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Copying Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

Creating Revision Labels

Like view labels, new revision labels can be created from the **View** menu. In fact, if you are creating a new revision label based on an existing revision label in another view, you must use the **View** menu for this purpose. However, revision labels can also be created from the **Folder Tree** menu, a component menu, or the context menu.

To create a new revision label for selected items (from the View menu)

- 1 Select a folder in the folder hierarchy tree.
- 2 Select one or more items on any of the tabs in the upper pane.
- 3 Choose **View** ▸ **Labels** and click the **Revision** tab in the **Labels** dialog box.
The labels are listed in reverse chronological order based on the time at which they were created.
- 4 Type a name and description for the label.
The maximum label name length is 64 characters and the description length is 254 characters.
- 5 Optionally, check **Frozen** to freeze the label so that revisions attached to it cannot be changed.
- 6 Click **OK**.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)
[Attaching Labels to Folders](#)
[Creating View Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Deleting Labels](#)
[Detaching Labels from Items](#)

Creating View Labels

View labels, usually used as build labels by default, can be extremely useful when you want to label every folder and item in a particular view.

To create a view label

- 1 Open the view to which you want to apply the label.
- 2 Choose **View ▸ Labels**.
The **Labels** dialog box opens with the **View** tab already selected. This tab lists existing view labels in reverse chronological order, based on the time when they were created.
- 3 Click **New** to create a new label and add its name to the list box.
The **View Label** dialog box opens.
- 4 Type a name and description for the label in the appropriate text boxes. The maximum label name length is 64 characters, and the description length is 254 characters.
- 5 Select one of the following options:
 - ◆ **Current Configuration**: This attaches the label to the tip revision of every item in this view's current configuration.
 - ◆ **Labeled Configuration**: This option attaches the label to the revisions of items to which the label you specify is currently attached. (Creating a view label based on another view label is equivalent to copying that view label.)
 - ◆ **Promotion State Configuration**: This option attaches the label to the revisions of items currently in the promotion state that you specify. (Actually, the label is attached to the revision that has the promotion state's current view label.)
 - ◆ **Configuration As Of**: This option attaches the label to the revision of every item that was a tip revision at the specified date and time.
- 6 Optionally, check **Use As Build Label** to update each change request that has **Next Build** as the setting for its **Addressed In Build** property. If this option is not selected, the view label will still be attached to change requests, but the setting of the **Addressed In Build** property will not change.
- 7 Optionally, to freeze the label so that the revisions attached to it cannot be changed, check **Freeze**.
- 8 Click **OK**.

Note: It is always important to synchronize the dates and times of the computers that run the StarTeam clients and the StarTeam server. However, if they are not synchronized and you select the current time as a label's configuration, the label may not be immediately visible.

Related Concepts

[Labels](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Creating Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

Deleting Labels

In StarTeam, you can completely remove a view or revision label from a view, although you can create a new label with the same name later, if desired. When you delete a label, it is no longer visible in any list of labels, nor is it attached to any folders or items.

Note: If a label is frozen, you must unfreeze it before you can delete it.

To delete a view label

- 1 Open the view for which you want to delete a label.
- 2 Choose **View** ► **Labels**.
This opens the **Labels** dialog box.
- 3 Select the tab for the type of label you are deleting, the **View** tab for a view label, or the **Revision** tab for a revision label.
These tabs list existing labels in reverse chronological order, based on the time that they were created.
- 4 Select the label and click **Delete**.
This removes the label from the view.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)
[Creating View Labels](#)
[Creating Revision Labels](#)
[Reviewing and Moving Labels](#)
[Copying Revision Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Detaching Labels from Items](#)

Demoting View Labels

Sometimes a labeled set of files is promoted prematurely and must be demoted. For example, if a specific build is promoted to the Beta state, but contains serious flaws, it should probably be returned to the prior promotion state. You can only demote view labels by editing the promotion state.

To demote a view label to a previous state

- 1 Choose **View** ► **Promotion** to display the **Promotion** dialog box.
- 2 Click Edit to open the **Promotion State** dialog box.
- 3 Select a different view label from the **View Label** drop-down list.
- 4 Click **OK**.

Related Concepts

[Overview of Views](#)

[Labels](#)

[Folders](#)

[Understanding View Types](#)

[Proper Use of Views](#)

[Understanding Branching](#)

Related Procedures

[Rolling Back the Current View Configuration](#)

[Promoting View Labels](#)

[Managing Views](#)

[Creating and Configuring Views](#)

[Creating View Labels](#)

[Copying View Labels](#)

[Creating Revision Labels](#)

[Copying Revision Labels](#)

[Reviewing and Moving Labels](#)

[Freezing or Unfreezing Labels](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Detaching a Label from a Rolled-back View

Sometimes you need to detach a label from an item in a rolled-back view. For example, suppose you deleted a file that had view labels attached to it. Later you created a build label based on one of the view labels that was attached to the deleted file. If you roll back the view to the new build label in order to perform a build, the deleted file reappears in your view. If you do not want that file in this build, you can detach the new build label from that file. If you try to detach any other label from the rolled-back view, an error message informs you that you can detach only the label to which the view has been rolled back.

Note: You can attach and detach any labels from items in current view configurations, but you cannot see deleted items in those configurations. You can detach view labels from deleted items only if you roll back the view to a configuration based on the label you want to detach.

To detach a view label from a in a rolled-back view

- 1 Roll back the view to the view label you want to detach.
- 2 From the upper pane, select the item from which the view label is to be detached.
- 3 Click the **Label** tab in the lower pane.
- 4 Right-click the label you wish to remove and choose **Detach**.
- 5 Click **OK**.

The item from which the label is detached will disappear after a refresh.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Detaching Labels from Folders](#)
[Detaching Labels from Items](#)
[Attaching Labels to Items](#)
[Creating View Labels](#)
[Creating Revision Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Deleting Labels](#)

Detaching a Label from a Specific Revision

If you decide not to include certain items in a view or revision label, you can detach the label from those items individually or as a group. Generally, the items from which labels are detached are files or folders.

To detach a label from a specific item revision

- 1 Select a folder from the folder hierarchy.
- 2 Click the component tab in the upper pane containing the item you want to detach, and select the item.
- 3 Click the **Label** tab in the lower pane.
- 4 Double-click a revision in the **Label** pane to see all labels attached to it as children of the revision.
- 5 Right-click the label you wish to remove and choose **Detach**.
- 6 Click **OK**.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Copying Revision Labels](#)

[Reviewing and Moving Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

Detaching Labels from Folders

If you decide not to include certain folders in a view or revision label, you can detach the label from those folders.

Note: You can attach and detach any labels from items in current view configurations, but you cannot see deleted items in those configurations. You can detach view labels from deleted items only if you roll back the view to a configuration based on the label you want to detach.

To detach a view or revision label from a folder and its contents

- 1 Right-click the folder in the **Folder Tree** or on the **Folder** tab and choose **Folder** ► **Labels** to open the **Labels** dialog box.
This dialog box lists the labels currently attached to this folder.
- 2 Select the label to be detached from the folder.
- 3 Click **Detach**.
- 4 Optionally, if you are detaching a revision label, select the **Folder Only**, **Folder And Items Contained In Folder**, or **Everything In Subtree Rooted At Folder** option.

Note: When you detach a view label from a folder, the label is automatically detached from the items that the folder contains. It is also automatically detached from the child folders and their contents.

The folder from which the label is detached will disappear after a refresh.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Detaching Labels from Items](#)
[Detaching a Label from a Rolled-back View](#)
[Attaching Labels to Items](#)
[Creating View Labels](#)
[Creating Revision Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Deleting Labels](#)

Detaching Labels from Items

If you decide not to include certain items in a view or revision label, you can detach the label from those items individually or as a group. Generally, labels are detached from files or folders.

Note: You can attach and detach any labels from items in current view configurations, but you cannot see deleted items in those configurations. You can detach view labels from deleted items only if you roll back the view to a configuration based on the label you want to detach.

To detach a view or revision label from selected items

- 1 Select a folder from the folder hierarchy tree.
- 2 Click the component tab containing the items you want to detach.
- 3 Select the items that do not require the label.
- 4 Right-click the selected items and choose **Labels** ► **Detach**.

The **Detach a Label** dialog box opens and displays all existing labels, identifying them as view or revision labels. By default, both **View Labels** and **Revision Labels** are checked.

- 5 Optionally, uncheck either **View Labels** or **Revision Labels** to limit the display list to a specific type of label.
- 6 Select a label from the list, and click **OK**.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Detaching Labels from Folders](#)
[Detaching a Label from a Rolled-back View](#)
[Attaching Labels to Items](#)
[Creating View Labels](#)
[Creating Revision Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Deleting Labels](#)

Freezing or Unfreezing Labels

When a label is frozen, the label cannot be:

- ◆ Attached to any additional folders or items.
- ◆ Detached from any folder or items.
- ◆ Moved from one revision of a folder or item to another.

Tip: You can identify a frozen label by a label icon containing a small snowflake displaying on a round, blue background. The icon displays in front of the label name in the list box.

To freeze or unfreeze a label

- 1 Choose **View** ▸ **Labels** to open the **Labels** dialog box.
- 2 Do one of the following:
 - ◆ Click the **View** tab if the label to be frozen is a view label.
 - ◆ Click the **Revision** tab if the label to be frozen is a revision label.

Both tabs list existing labels in reverse chronological order, based on the time they were created.

- 3 Select the label from the list box.
- 4 Click **Freeze** or **Unfreeze**.
- 5 Click **Close** when your are finished.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Folders](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Copying Revision Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Promoting View Labels

You can promote a view label from one promotion state to the next if you have the appropriate access rights.

To promote a view label to the next promotion state

- 1 Choose **View** ► **Promotion** to open the **Promotion** dialog box which displays any states currently created for the view.
The states are displayed from the final state down to the initial state.
- 2 Select the promotion state currently associated with the view label that you want to promote.
- 3 Click **Promote**. The **Promote View Label** dialog box indicates that the view label is now associated with the next state (the state immediately above the selected state in the **Promotion** dialog box).
- 4 Verify that the information is what you were expecting to see, then click **OK**.

The selected view label now applies to two promotion states: the one to which it was promoted and the one you originally selected. Usually, your next action is to associate a new view label with the original state.

Related Concepts

[Overview of Views](#)
[Labels](#)
[Folders](#)
[Understanding View Types](#)
[Proper Use of Views](#)
[Understanding Branching](#)

Related Procedures

[Rolling Back the Current View Configuration](#)
[Demoting View Labels](#)
[Managing Views](#)
[Creating and Configuring Views](#)
[Creating View Labels](#)
[Copying View Labels](#)
[Creating Revision Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Demoting View Labels](#)
[Deleting Labels](#)
[Detaching Labels from Items](#)

Reviewing and Moving Labels

To see the labels currently attached to specific items and folders, you can display them on the **Label** pane. In the case of revision labels, you can also move them to another revision of the folder or item.

To review all labels attached to item revisions

- 1 Select a folder in the folder hierarchy tree.
- 2 Click a component tab in the upper pane and select an item.
- 3 Click the **Label** tab in the lower pane.
This displays the **Label** pane which shows all revisions for the item.
- 4 Double-click a revision to expand the revision node and display its labels.

To move a revision label from one item revision to another

- 1 Select a folder in the folder hierarchy tree.
- 2 Click a component tab in the upper pane and select an item.
- 3 Click the **Label** tab in the lower pane.
This displays the **Label** pane which shows all revisions for the item.
- 4 Double-click a revision to expand the revision node and display its labels.
- 5 Select a revision label and drag it from one revision to another.

Note: A label can be attached to only one revision.

To review all labels attached to a folder's revisions

- 1 Select a folder in the folder hierarchy tree.
- 2 Choose **Folder Tree** ▶ **Labels** to open the **Labels** dialog box.
This displays all labels currently attached to this folder on a revision-by-revision basis.

To move a revision label from one folder revision to another

- 1 Select a folder in the folder hierarchy tree.
- 2 Choose **Folder Tree** ▶ **Labels** to open the **Labels** dialog box.
This displays all labels currently attached to this folder on a revision-by-revision basis.
- 3 Drag a revision label from one folder revision node to another.

Note: A label can be attached to only one revision.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Copying Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Sorting Labels Alphabetically

This topic describes how to sort labels alphabetically in descending or ascending order by the column name in the **Attach a Label** dialog box. It also explains how to change the ordering of the columns within the dialog box. The **Attach a Label** dialog box provide the following columns, all sortable and moveable for both view and revision labels:

- ◆ Name
- ◆ Description
- ◆ Time
- ◆ Creation On
- ◆ Created By

Note: The sorting and repositioning selections for the columns that you choose in the **Attach a Label** dialog box do not persist between sessions.

To sort labels alphabetically for Folders

- 1 Right-click a folder in the **Folder Tree** and choose **Labels** to open the **Labels** dialog box.
- 2 Click **Attach** to open the **Attach a Label** dialog box.
- 3 Click a desired column heading in the **Attach a Label** dialog box to sort alphabetically. Click the column heading again to reverse the order.

To sort labels alphabetically for items

- 1 Right-click an item in the upper pane and choose **Labels** ► **Attach**.
This opens the **Attach a Label** dialog box.
- 2 Click a desired column heading in the **Attach a Label** dialog box to sort alphabetically. Click the column heading again to reverse the order.

To reposition columns in the Attach A Label and Detach A Label dialog boxes

- 1 Open the **Attach a Label** dialog box as described in one of the previous procedures.
- 2 Click a column heading and drag it to reorder its location within the dialog box.

Related Concepts

[Folders](#)
[Labels](#)
[Attaching Labels to Items](#)

Related Procedures

[Creating Revision Labels](#)
[Detaching Labels from Items](#)
[Reviewing and Moving Labels](#)
[Copying Revision Labels](#)

Linking and Unlinking Items

This section contains procedures related to linking and unlinking items.

In This Section

[Checking Linked Files In and Out](#)

Describes how to check linked files in and out from the linked item.

[Creating External Links](#)

Describes how to link folders and items across different servers.

[Customizing Link Item Properties](#)

Describes how to modify item properties from a link.

[Customizing Link Properties](#)

Describes how to modify link properties.

[Deleting Links](#)

Describes how to delete a link.

[Displaying Only Enhanced Process Links](#)

Describes how to display enhanced process links in the Link pane.

[Enabling the use of Enhanced Process Links](#)

Describes how to enable the use of enhanced process links in the client for a project.

[Filtering Process Tasks From Other Tasks](#)

Describes how to filter process tasks from other tasks.

[Finding Files Linked to Active Process Items](#)

Describes how to find all the files linked to an active process item.

[Linking Files to Process Items](#)

Describes the process of linking and pinning files to process items and to active process items..

[Linking Items Internally or Externally](#)

Describes how to link folders and items both internally in the same view, or externally between items on two different servers or configurations.

[Linking Specific Revisions](#)

Describes how to link to specific revisions.

[Reviewing Linked Change Requests](#)

Describes how to review linked change requests.

[Selecting Linked Files](#)

Describes how to select linked files from a link item.

[Viewing Links](#)

Describes how to view links in the Link pane.

Checking Linked Files In and Out

If you are checking in a file that has one or more linked change requests, you can do this from the **Link** pane.

To check in linked files

- 1 Click the component tab in the upper pane containing the item linked to the file(s) you want to check in or out.
- 2 Select the item.
- 3 Click the **Link** tab in the lower pane to display the list of files linked to this item.
- 4 Select one or more files in the **Link** pane.
- 5 Right-click the selected item in the upper pane or the selected files in the lower pane.
- 6 Do one of the following:
 - ◆ Choose **Linked Files** ► **Check In All**.
 - ◆ Choose **Linked Files** ► **Check Out All**
- 7 Use the **Check In** or **Check Out** dialog box as you normally would to check files in our out.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)
[Linking Files to Process Items](#)
[Checking In Files](#)
[Checking Out Files](#)

Creating External Links

A *link* is a connection between two folders, two items, or a folder and an item. An external link links an item on one server to an item on another server.

Creating links can be quite useful. For example, linking a file to a change request allows you to mark it as fixed when you check in the edited file. By linking files to the requirements document that the files fulfill, you can easily refer to or update the document.

Note: To create external links, you must have the access rights to the generic external link access rights, such as create, see, modify, delete.

To create an external link

- 1 Open both projects and servers which have items you want to link. External links will not work unless both servers are opened in StarTeam before you create the link.
- 2 Begin the link process by doing one of the following:
 - ◆ Select a folder in the folder hierarchy or in the upper pane on the **Folder** tab.
 - ◆ Click a component tab in the upper pane, such as **File**, **Change Request**, **Requirement**, **Topic**, or **Task**, and select one or more items.
- 3 Right-click the selected item(s) on the component tab and choose **Links** ► **Create Link**.
The **Links** menu is also available on the component menu that corresponds with the selected component tab.
This action changes the mouse pointer and displays it as a knotted rope.

Note: If you initially select an item from the upper pane, you can also use the **Link** button on the toolbar; however, this button is disabled if you start the link with a folder.

- 4 Select the folder or item(s) for the end of the link in the project on the other server. This can be:
 - ◆ A StarTeam folder (if you have not already selected a folder).
 - ◆ One or more other files.
 - ◆ One or more change requests.
 - ◆ One or more requirements.
 - ◆ One or more topics and/or responses.
 - ◆ One or more tasks and/or subtasks.

To locate all items, you may need to switch to a different component tab or use the **All Descendants** button on the toolbar.

- 5 Choose **Links** ► **Complete Links** on the **Folder** menu, the component menu, or the context menu, or click the **Link** button again on the toolbar.
This button is disabled if you are linking an item to a folder.
- 6 Verify that the links exist by doing one of the following:
 - ◆ Select a linked item, then click the **Link** tab on the lower pane to view the links for the item.

- ◆ Right-click a linked folder, then choose **Properties** to display the **Folder Properties** dialog box. Click the **Link** tab to view the link. (The **Link** tab will not appear in this dialog box if you do not have access rights to view links.)

You can also view a link by selecting either of its ends. The end you select, whether a folder or an item, is called the source. The other end of the link is called the target and is listed in the **Item Type** column on the **Link** pane.

Tip: If you change your mind about creating a link after you have started to create it, but before you have finished completing it, you can select **Links** ► **Cancel Link** on the **Folder Tree** menu, the component menu, or the context menu. If you are using the **Link** button on the toolbar, press **ESC**.

Note: External links can also be created using drag-drop. With both views open, select the source item, press **CTR** + **SHIFT**, then drag-drop it on the target item.

Related Reference

[External Link Access Rights](#)

Customizing Link Item Properties

You can view or modify folder and item properties directly from the **Link** pane.

To view or modify folder or item properties from a link

- 1 Select an item in the upper pane that links to a folder or another item.
- 2 Select the **Link** tab in the lower pane.
- 3 Right-click a link and choose **Item Properties** to display a **Properties** dialog box.
This dialog box displays information about the folder or item in the **Item** column in the **Link** pane.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Related Reference

[Project, Folder, and Item Properties](#)

Customizing Link Properties

You can view or modify link properties from the **Link** pane.

To view or modify link properties from a folder

- 1 Right-click the folder and choose [Properties](#).
- 2 Click the **Link** tab in the **Folder Properties** dialog box.
The **Link** pane displays all links for the selected folder.
- 3 Right-click a link and choose [Link Properties](#).
The **Link Properties** dialog box opens where you can view or modify certain properties or add a comment.

To view or modify link properties from an item

- 1 Select an item on one of the component tabs in the upper pane.
- 2 Select the **Link** tab in the lower pane.
The **Link** pane displays all links for the selected item.
- 3 Right-click a link in the **Link** pane and choose [Link Properties](#).
The **Link Properties** dialog box opens where you can view or modify certain properties or add a comment.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Related Reference

[Project, Folder, and Item Properties](#)

Deleting Links

You can delete an existing link from either the source end or the target end of the link.

To delete a link

- 1 Select the folder or item in the upper pane that is the source or target of the link.
- 2 Click the **Link** tab in the lower pane and select one or more links to delete.
- 3 Right-click the selected links and choose **Links** ► **Delete Link**.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Linking Files to Process Items](#)

Displaying Only Enhanced Process Links

If your project has enhanced process links enabled, you can choose to display only enhanced process links in the **Link** pane.

To display only enhanced process links in the Link pane

- 1 Click the **Link** tab at the bottom of the client.
- 2 Select the radio button **Show enhanced links only** at the top of the **Link** pane.

Now when you select an item in the upper pane that has links, the **Link** pane only shows enhanced links, not standard links.

Related Concepts

[Overview of Projects](#)

[Process Items and Process Links](#)

Related Procedures

[Enabling the use of Enhanced Process Links](#)

Enabling the use of Enhanced Process Links

In the Cross-platform Client, each user can specify that a project uses the enhanced process link model instead of the standard linking model. In the standard linking model, if a given item is specified as the reason for a change, then process links are created directly from that process item to each changed file and folder.

In the enhanced linking model, the process item (the item specified as the reason for making a given set of changes) is distinguished from the process task that is automatically created to represent the associated changes in a particular view. Changed files and folders are linked to the process item indirectly, through a process task.

Note: Borland recommends that you either specify both the **Enable enhanced links for all projects** and the **Enable enhanced links for new projects** options on the **General** tab in the **Configure Server** dialog box, or leave both options deactivated for the specified server configuration. You can then control the usage of enhanced process links on a per-project-basis in the **Process Rules** tab of Cross-Platform Client's **Project Properties** dialog box. For example, if you activate both options for the server configuration, then you could turn off this option in the **Process Tab** of the Cross-Platform Client's **Project Properties** dialog box by clearing the option to **Enable enhanced process links**. Once you start controlling these options per project in the **Project Properties** dialog box, the **Enable enhanced links for all projects** check box is no longer activated for the server configuration.

To enable the use of enhanced process links for a project in the client

- 1 Choose **Project** ► **Properties** to open the **Project Properties** dialog box.
- 2 Select the **Process Rules** tab.
- 3 Check **Enable Enhanced Process Links**, then click **OK**.

Related Concepts

[Overview of Projects](#)

[Process Items and Process Links](#)

Related Procedures

[Creating Projects](#)

[Configuring Server Configurations to use Enhanced Process Links](#)

Filtering Process Tasks From Other Tasks

If you have enabled enhanced process links in your project so that StarTeam is creating process tasks, you can filter your tasks to separate the process tasks from the regular tasks. Use the **Usage** field value to determine the difference between process tasks and standard tasks. If the **Usage** value is anything besides **Other**, then it is a process task.

To filter process tasks from other tasks

- 1 Choose **Task** ► **Filters** ► **Filters**.

This displays the **Filters** dialog box.

- 2 Click **New** and give the new filter a name.

Alternatively, copy an existing filter by selecting it, clicking **Save As**, and giving it a new name. Then select the copied filter and continue with the next steps.

- 3 Click **Fields** in the **Filters** dialog box.

- 4 Move the **Usage** field from the **Available Fields** list to the **Show Fields in this Order** list and click **OK**.

- 5 Click **Query** to open the **Queries** dialog box, and click **New**.

This opens the **Edit Query** dialog box.

- 6 Type a **Name** for the new query, and choose the following in the **Condition Node** section: **Field = Usage**, **Operator = Not Equal**, and **Value = Other**.

- 7 Click **Add** to add the condition to the query, and **Save** to save the query and return to the **Queries** dialog box.

Your new query is now highlighted in the list of queries.

- 8 Click **Select** in the **Queries** dialog box use this query in your new filter.

You are returned to the **Filters** dialog box, and your new filter should be highlighted.

- 9 Click **Save As** to save the filter.

Notice your new filter has been added to the **Filter** drop-down list at the top of the client.

To use the filter, simply select it from the **Filter** list on the **Task** tab.

Note: Conversely, you can create a filter that displays only the standard tasks. In the query, use the condition **Usage Equals Other**.

Tip: If you use tasks on a regular basis, and not just for process tasks, add **Usage Equals Other** to existing queries so you never see process tasks when working on tasks that have been manually created, or imported from Microsoft Project using Borland's Microsoft Project integration.

Related Concepts

[Process Items and Process Links](#)

[Process Tasks and Enhanced Process Links](#)

Related Procedures

[Creating Filters](#)

[Creating Queries](#)

Finding Files Linked to Active Process Items

When you have files linked to an active process item, you can quickly find all the files linked to it.

To find files linked to an active process item

- 1 Click the **Change Request**, **Requirement**, or **Task** tab in the upper pane that contains the active process item.
You can see what item is the active process item by looking at the left side of the **Status Bar**. The second box in the **Status Bar** displays the **Active Process Item** icon, followed by the name of the item.
- 2 Select the active process item and click the **Link** tab in the lower pane.

The **Link** pane displays all the files linked to that item.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)

Linking Files to Process Items

If process rules are enforced for a project, linking and pinning new file revisions to a process item is required. Otherwise, this step is optional, and you can select any change request, requirement, or task as a process item.

To link and pin a file revision to a process item

- 1 Click the **File** tab in the upper pane and select a file or files to be checked in.
- 2 Choose **File** ► **Check In** to display the **Check In** dialog box.
- 3 Check **Link and Pin Process Item**.
- 4 Click **Select** to open the **Select Process Item** dialog box.
- 5 Select one of the following to limit the list of possible process items:
 - ◆ **List All Permitted Items**: Displays all items that can be used as process items. If process rules are not enforced, the list contains all change requests, requirements, and tasks.
 - ◆ **List All Permitted Items Assigned To Me**: Displays all the items for which you are responsible that can be used as process items.
 - ◆ **List Linked Items**: Lists the process items that are already linked to at least one of the files you are checking in. No process item appears on the list more than once, even if it is linked to several files. Also, when a process item is linked to more than one file, the dialog box displays the name of only one file. Despite this fact, the application will update or create links for every file being added.
- 6 Select the **Change Request**, **Requirement**, or **Task** tab to restrict the list to a specific type of item.
- 7 Select one item in the list to be the active process item and click **OK**.
- 8 Optionally, in the **Check In** dialog box, check **Mark Selected Item As Fixed/Finished/Complete** if work on the process item is completed.
- 9 Complete filling in the fields in the **Check In** dialog box and press **OK**.

Tip: If process rules are enforced, the use of some change requests, requirements, or tasks as process items may not be permitted because of their status. If you select such an item and click **OK**, the application notifies you of this fact. By double-clicking the item in the list box, you can display its properties and change it to a permitted status for a linked process item.

To link and pin a file revision to the active process item

- 1 Click the **File** tab in the upper pane and select a file or files to be checked in.
- 2 Choose **File** ► **Check In** to display the **Check In** dialog box.
If you have already specified a process item, it will appear by default in the **Link and Pin Process Item** field.
- 3 Check **Link and Pin Process Item** to accept this active process item.
- 4 Optionally, in the **Check In** dialog box, check **Mark Selected Item As Fixed/Finished/Complete** if work on the active process item is completed.
- 5 Complete filling in the fields in the **Check In** dialog box and press **OK**.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)

Linking Items Internally or Externally

This procedure describes how to link two items, either internally in the same server configuration, or linking between two items located on different server configurations, called *external linking*.

In StarTeam, an *item* is a file, change request, requirement, task, or topic. A *link* is a connection between two folders, two items, or a folder and an item. Creating links can be quite useful. For example, linking a file to a change request allows you to mark it as fixed when you check in the edited file. By linking files to the requirements document that the files fulfill, you can easily refer to or update the document.

You can create several links at the same time if you want to link several items of the same type to one particular item. For example, you might wish to link several change requests to a single file. To accomplish this, you can create links using the **Folder Tree** menu, component menu, context menu, or **Link** button on the toolbar.

Note: When you create external links between items on different server configurations, both server configurations need to be opened in the Cross-Platform Client to be able to create or view the external links.

To link a folder or item(s) to one or more items

- 1 Begin the link process by doing one of the following:
 - ◆ Open the server configuration and view which contains the items you want to link. If you want to link two items on different server configurations, open both server configurations and views in the Cross-Platform Client.
 - ◆ Select a folder in the folder hierarchy or in the upper pane on the **Folder** tab.
 - ◆ Click a component tab in the upper pane, such as **File**, **Change Request**, **Requirement**, **Topic**, or **Task**, and select one or more items.

Note: You cannot link two folders to each other with the method described in this procedure.

- 2 Right-click the selected item(s) on the component tab and choose **Links** ► **Create Link**.
The **Links** menu is also available on the component menu that corresponds with the selected component tab.
This action changes the mouse pointer and displays it as a knotted rope.

Note: If you initially select an item from the upper pane, you can also use the **Link** button on the toolbar; however, this button is disabled if you start the link with a folder.

- 3 Select the folder or item(s) for the end of the link from the same configuration or from the external server configuration. This can be:
 - ◆ A StarTeam folder (if you have not already selected a folder).
 - ◆ One or more other files.
 - ◆ One or more change requests.
 - ◆ One or more requirements.
 - ◆ One or more topics and/or responses.
 - ◆ One or more tasks and/or subtasks.

To locate all items, you may need to switch to a different component tab or use the **All Descendants** button on the toolbar.

- 4 Choose **Links** ► **Complete Links** on the **Folder** menu, the component menu, or the context menu, or click the **Link** button again on the toolbar.

This button is disabled if you are linking an item to a folder.

- 5 Verify that the links exist by doing one of the following:

- ◆ Select a linked item, then click the **Link** tab on the lower pane to view the links for the item.
- ◆ Right-click a linked folder, then choose **Properties** to display the **Folder Properties** dialog box. Click the **Link** tab to view the link. (The **Link** tab will not appear in this dialog box if you do not have access rights to view links.)

You can also view a link by selecting either of its ends. The end you select, whether a folder or an item, is called the source. The other end of the link is called the target and is listed in the **Item Type** column on the **Link** pane.

Tip: If you change your mind about creating a link after you have started to create it, but before you have finished completing it, you can select **Links** ► **Cancel Link** on the **Folder Tree** menu, the component menu, or the context menu. If you are using the **Link** button on the toolbar, press **ESC**.

Related Concepts

[Links: Internal and External](#)

Related Procedures

[Linking Specific Revisions](#)

[Checking Linked Files In and Out](#)

[Selecting Linked Files](#)

[Deleting Links](#)

[Linking Files to Process Items](#)

[Creating External Links](#)

Linking Specific Revisions

Each end of a link has an associated start revision and an end revision that determines the range of revisions to which the link applies. The start revision is always fixed at the time of the creation of the link and is set to the first revision on the current branch. The end revision is under the user's control and may be fixed (or pinned), which puts an upper bound on the linked revisions, or floating, which does not. If a link end is pinned, it is always attached to the same version of the linked folder or file. If a link end floats, it moves from revision to revision, as new revisions of the linked folder or item are created.

By default, a link connects the tip revisions of the linked pair. The revisions selected for both links appear as columns on the **Link** pane.

Determining whether a link is visible on a given item is simple. If any of the revisions between the start and the end revision defined for the link are in the history of the selected item, it is visible. Otherwise, it is not.

To link to a tip revision

- 1 Select a folder or item for which you have created a link.
- 2 Do one of the following:
 - ◆ If you selected an item in the upper pane, click the **Link** tab in the lower pane.
 - ◆ If you selected a folder in the **Folder Tree**, choose **Folder Tree** ► **Properties** to display the **Folder Properties** dialog box, and click the **Link** tab.
- 3 Select one or more links in the **Link** pane.
- 4 Right-click the selected link(s) and choose one of the following options:
 - ◆ **Pin Link** ► **To Source Item At Tip**: This command pins the link to the tip revision of the source (that is, the folder whose properties you are reviewing or the item selected from the upper pane).
 - ◆ **Pin Link** ► **To Target Items At Tip**: This command pins the link to the tip revision of the target (that is, the folder or item in the Item column on the Link pane).
 - ◆ **Pin Link** ► **To Source And Target Items At Tip**: This command pins the link to the tip revisions of both the source and target.
 - ◆ **Float Link** ► **To Source Item**: This command allows the link to float from tip revision to tip revision of the source as new revisions are created.
 - ◆ **Float Link** ► **To Target Items**: This command allows the link to float from tip revision to tip revision of the target as new revisions are created.
 - ◆ **Float Link** ► **To Source And Target Items**: This command allows both the source and the target of the link to float from tip revision to tip revision.

To link to a specific revision

- 1 Right-click a folder or item for which you have created a link.
- 2 Do one of the following.
 - ◆ If you selected an item, click the **Link** tab in the lower pane.
 - ◆ If you selected a folder in the **Folder Tree**, choose **Folder Tree** ► **Properties** to display the **Folder Properties** dialog box, and click the **Link** tab.
- 3 Select a link.

- 4 Right-click the selected link and choose [Link Properties](#).
The **Link Properties** dialog box opens.
- 5 Optionally, type a description or comment about the link in the **Comment** text box.
This text will appear in the **Comment** column of the **Link** pane.
- 6 Do one of the following in the **Source Item** group box:
 - ◆ Click **Pin** to display the **Select Version** dialog box. Select a specific folder or item revision from the list. This revision number will appear in the **Selection Version** column of the **Link** pane.
 - ◆ Click **Float** so that the link is always connected to the tip revision of this item.
- 7 Do one of the following in the **Is Linked to Target Item** group box:
 - ◆ Click **Pin** to display the **Item Version** dialog box. Select a specific folder or item revision from the list. This revision number will appear in the **Item Version** column of the **Link** pane.
 - ◆ Click **Float** so that the link is always connected to the tip revision of this item.
- 8 Click **OK**.

Note: You can link items from a project view on one server to an item in another project or view on a different server. This is called an external link.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Reviewing Linked Change Requests

If you are checking in a file that has one or more linked change requests, you should also review all change requests associated with the file.

To review a linked change request

- 1 Select the file in the upper pane that is linked to the change request.
- 2 Choose **File** ► **Check In**.
This opens the **Check In** dialog box.
- 3 Click **Advanced** to open the **Advanced Options** dialog box.
- 4 Click **Show Change Requests**.

The **Advanced Options** dialog box expands at the bottom and displays the list **Change Requests Linked In This View**.

Note: No change request appears in the list more than once, even if it is linked to several of the files you are checking in. When a change request is linked to more than one file, the list displays the name of only one of the files.

- 5 Optionally, double-click a change request to review or edit its properties.
- 6 Optionally, check **Marked Selected Change Requests As Fixed**.
If you check this option, StarTeam will mark the selected, but unresolved, change request **Fixed** as part of the check-in process.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Selecting Linked Files

You can quickly select all the files associated with a linked item directly from the component tab in the upper pane. There are two choices for selecting the linked files:

- ◆ You can select only the files linked to an item.
- ◆ You can add the files linked an item to an existing selection of files on the **File** pane.

To select all the files linked to a specific item

- 1 Click the component tab in the upper pane containing the item to which the files are linked, and select the item.
- 2 Right-click the selected item and choose **Linked Files** ► **Add To Selection**.

StarTeam switches the upper pane to the **File** pane, activates the **All Descendants View**, and adds all the files linked to the item to the existing file selection.

Note: Selected items must have linked files in the current view to perform this operation.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Customizing Link Properties](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Viewing Links

StarTeam displays links at the bottom of the client in the **Link** pane for any item selected item in upper pane.

To view links

- 1 Select an item in the upper pane.
- 2 Click the **Link** tab at the bottom of the client.

You can control whether to display all links, or only enhanced links by selecting the appropriate option at the top of the **Link** pane.

You can also use the drop-down list box in the **Link** pane (called **Linked Item**) that identifies the item for which links are displayed. Initially, the item displayed is the one selected in the upper pane. To select a different linked item, click the arrow beside the **Linked Item** list box

Note: You can drag any link displayed in the Link tab to the **Linked Item** list box. This action changes the linked item as well as the links displayed in the **Link** pane. This feature is especially helpful in viewing the links to process tasks. If you switch to a different tab and then return, the list box reverts to displaying only the item selected from the upper pane.

When you select a process item in the upper pane, the only link displayed in the **Link** pane may be the link to a process task. To see the items that are linked to the process task (and are therefore linked indirectly to your process item), drag the process task link from the list in the **Link** pane onto the **Linked Item** list box. The **Linked Item** list box now displays the process task, and the **Link** pane list now displays the process task links. For example, the **Link** pane could show a link to each file checked in using the previously selected process item, as well as a link to the process item itself.

Related Concepts

[Overview of Projects](#)

[Process Items and Process Links](#)

Related Procedures

[Enabling the use of Enhanced Process Links](#)

Querying Data

You can build and run queries to limit the data you view in the various StarTeam components in the upper pane. Queries are more complex than filters which control the arrangement of the information presented.

In This Section

[Applying Queries](#)

Describes how to apply an existing query.

[Copying Queries](#)

Describes how to copy queries.

[Creating Queries](#)

Describes how to create both simple and complex queries.

[Creating “Me” Queries](#)

Describes how to create a query that displays only the items created by the person logged into the computer as the user.

[Deleting Queries](#)

Describes how to delete queries.

[Editing Queries](#)

Describes how to edit a query.

[Selecting Change Requests Using a Query](#)

Describes how to select change requests using a query.

Applying Queries

Once a query has been created, you can use it to:

- ◆ List only those items that match the query in the upper pane.
- ◆ Select items from the upper pane that match the query (even though other items are still displayed).

Note: Using a query allows you to treat items as a group. For example, you can print a report on them, attach a label to them, check them out, and so on.

To apply an existing query to the items in the upper pane

- 1 Select a folder from the folder hierarchy.
- 2 Click the component tab on which you want to run the query.
- 3 Do one of the following:
 - ◆ Right-click a column header on upper pane and choose **Queries** from the context menu.
 - ◆ Choose **Filters** ► **Queries** from the component menu.

The **Queries** dialog box opens.

- 4 Select a query in the list box.
- 5 Click **Select** to apply the query to the items shown on the upper pane. This action changes the contents of the upper pane by displaying only those items that meet the specifications of the query.
- 6 If you are not satisfied with the results of the query, reopen the **Queries** dialog box, and click **Clear Query** to return to the previously displayed list of items.

Tip: You can also edit the query or create a new query to get the results you want.

To select the items in the upper pane that match an existing query

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Choose **Select** ► **By Query** from the component or context menu. The **Select Query** dialog box lists all existing queries.
- 4 Select the query of interest.
- 5 Click **OK**. This action highlights the items in the upper pane that match this query.
- 6 To deselect the query items, simply click the upper pane.

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

Copying Queries

StarTeam allows you to create new queries quickly by copying an existing query and editing it. Using this feature saves time because you do not have to recreate the query conditions.

To copy a query

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Do one of the following:
 - ◆ Right-click a column header on upper pane and choose **Queries** from the context menu.
 - ◆ Choose **Filters** ► **Queries** from the component menu.

The **Queries** dialog box opens.

- 4 Select the query to be edited from the list, and then click **Copy**. The **Copy Query** dialog box opens.

Tip: Public queries have a multiuser icon to the left of the query name; private queries have a single-user icon.

- 5 Do the following:
 - ◆ Type the name for the new query in the **Query Name** text box.
 - ◆ Select the **Public** check box to add this query to the project (and the server configuration), allowing anyone with the appropriate access rights to use it. If you do not check the **Public** check box, the query will be private – that is, available only to your user ID.
 - ◆ Click **OK**. The new query displays in the **Queries** dialog box.
- 6 To change the conditions in the query, select it from the **Queries** dialog box and click **Edit**. This action opens the **Edit Query** dialog box.
- 7 Edit the appropriate nodes of the tree.
- 8 Click **Save**. This action redisplay the **Queries** dialog box.

Note: If you do not have the access rights to create a public query for this project, you can create a private query.

- 9 Click **Close**.

Note: If this is a public query, you might want to set access rights for it.

Related Concepts

[Queries](#)

Related Procedures

[Editing Queries](#)

Creating Queries

You can write simple queries that have only one condition, or complex queries that use several conditions and one or more logical operators.

To create a query

- 1 Select a folder in the **Folder Tree**.
- 2 Select the component tab in the upper pane for the item type you want to query.
- 3 Right-click in the upper pane and choose **Filters** ► **Queries**.
This opens the **Queries** dialog box.
- 4 Click **New** to open the **New Query** dialog box.
- 5 Type a name for your query.
- 6 Decide whether you want this query to be public or private.
 - ◆ Leaving the **Public** checkbox empty means no one else can see this query but you.
 - ◆ Checking the **Public** checkbox adds this query to the project and the server configuration, and allows anyone with the appropriate access rights to use this query.
- 7 Select a **Field** and **Operator**, type or select a **Value**, and click **Add** to place this condition in the **Query** tree.
By default, the **Query** tree box contains the **AND** operator as the root of the tree, which you cannot delete. If there is only one condition, StarTeam ignores the logical operator.

Tip: Click **View as Text** to view the query in text format. Notice that the default logical **AND** operator was not included in your query. Click **OK** to close the window.

- 8 Optionally, click one of the following **Logical Node** buttons to create a new **Query** tree node: **AND**, **OR**, or **NOT**.

Tip: You can change an existing operator in a condition by toggling the **AND->OR->NOT** button. Keep clicking the button until the operator that appears is the one you want to use.

- 9 Select the fields for this new condition and click **Add**.

Note: A quick way to create a query that shows only the items

- 10 Add any other conditions, then click **Save**.

The **Queries** dialog box now contains your new query enabling you to select it for querying data.

Note: If this is a public query, you might want to grant access rights.

Tip: When creating a query condition, it is best to use the condition or logical operation that will result in the fewest matches as the first condition or logical operation.

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

[Creating “Me” Queries](#)

Creating “Me” Queries

StarTeam has the capability of creating "Me" queries that allows a query to be set up which is evaluated against the currently logged in user ("Me"), rather than having to specify a specific username at the time of query creation.

To create a “Me” query

- 1 Select a folder in the **Folder Tree**.
- 2 Select the component tab in the upper pane for the item type you want to query.
- 3 Right-click in the upper pane and choose **Filters** ► **Queries**.
This opens the **Queries** dialog box.
- 4 Click **New** to open the **New Query** dialog box.
- 5 Type a name for your query, for example “Me”.
- 6 Decide whether you want this query to be public or private.
 - ◆ Leaving the **Public** check box empty means no one else can see this query but the user logged into this system.
 - ◆ Checking the **Public** check box adds this query to the project and the server configuration, and allows anyone with the appropriate access rights to use this query.

Note: If this is a public query, you might want to grant access rights.

- 7 Select **Created By** for the **Field** in the **Condition Node** area. Select **Equals** for the **Operator**, and select 'Me' in the **Value** list. Click **Add** to place this condition in the **Query** tree.
By default, the **Query** tree box contains the **AND** operator as the root of the tree, which you cannot delete. If there is only one condition, StarTeam ignores the logical operator.
- 8 Click **Save**.
The **Queries** dialog box now contains your new query enabling you to select it for querying data.

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

[Creating Queries](#)

Deleting Queries

You can delete queries that you are sure you no longer use. Note that:

- ◆ You must have the appropriate access rights to delete a public query.
- ◆ You cannot delete a query that is referenced by a filter.

To delete an existing a query

- 1 Select a folder from the folder hierarchy and click the component tab with the query you want to delete.
- 2 Do one of the following:
 - ◆ Right-click a column header on upper pane and choose **Queries** from the context menu.
 - ◆ Choose **Filters** ► **Queries** from the component menu.

The **Queries** dialog box opens.

- 3 Select the query you want to delete.
- 4 Click **Delete**.
- 5 If you are sure that you want to delete the selected query, click **OK**. The query is removed from the **Queries** dialog box.
- 6 Click **Close**.

Related Concepts

[Queries](#)

Related Procedures

[Editing Queries](#)

Editing Queries

To display a useful set of data, you might need to edit or add to a query.

To edit a query

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Do one of the following:
 - ◆ Right-click a column header on upper pane and choose **Queries** from the context menu.
 - ◆ Choose **Filters** ► **Queries** from the component menu.

The **Queries** dialog box opens.

- 4 Select the query to be edited from the list, and then click **Edit**. The **Edit Query** dialog box opens.
- 5 Edit the appropriate nodes of the tree.
- 6 Click **Save**. This action re-displays the **Queries** dialog box, which lists the edited query.
- 7 Click **Close**.

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

Related Reference

[Queries](#)

Selecting Change Requests Using a Query

You can use a simple or complex query to limit the change requests displayed to those that fit specific criteria.

To select change requests using a query

- 1 Click **Change Request** tab in the upper pane.
- 2 Right-click a column header and choose **Queries** from the context menu.
- 3 Choose an existing query in the **Queries** dialog.
- 4 Click **Select** to apply it to the list of change requests.
- 5 Follow the steps in “Creating Queries” to define a new query.

Related Concepts

[Change Requests](#)
[Queries](#)

Related Procedures

[Customizing Change Request Reports](#)
[Creating Queries](#)
[Working with Change Requests](#)

Filtering Data

A filter lets you view data in a particular arrangement that consists of a set of fields (used as column headers), sorting and grouping information, and (usually) a query. Once a filter has been created, it can be used in every project that has the same server configuration.

In This Section

[Applying Predefined Filters](#)

Describes how to use the StarTeam predefined filters for grouping data.

[Copying Filters](#)

Describes how to copy an existing filter to use as the basis for a new filter.

[Creating Filters](#)

Describes how to create a filter from scratch or based on the current arrangement of data.

[Customizing Change Request Filters](#)

Describes creating a change request filter based on the current arrangement of data.

[Deleting Filters](#)

Describes how to delete a filter.

[Editing Filters](#)

Describes how to edit an existing filter.

[Filtering Process Tasks From Other Tasks](#)

Describes how to filter process tasks from other tasks.

[Resetting Filters](#)

Describes how to reset the filter as it was originally defined on the server.

[Showing Fields in a Change Request](#)

Describes how to show specific fields in a change request.

[Sorting and Grouping Change Requests](#)

Describes how to sort and group change requests.

[Sorting and Grouping Data](#)

Describes how to sort and group data.

Applying Predefined Filters

Existing public filters can be used on all projects in the same server configuration by any team members who have the appropriate access rights. Private filters can be used only by you.

To apply an existing filter

- 1 Select a folder from the folder hierarchy on the left.
- 2 Click a component tab in the upper pane.
- 3 Do one of the following:
 - ◆ Select a filter from the **Filters** drop-down list box on the toolbar. This action limits the data in the upper pane to the data that matches the filter.
 - ◆ Right-click a column header in the upper pane and choose **Filters**. Select a filter from the list of predefined filters in the **Filters** dialog box and click **OK**.

Related Concepts

[Filters](#)

Related Procedures

[Creating Filters](#)

Related Reference

[Filters Reference Topics](#)

Copying Filters

To save time, you can create a new filter by basing it on an existing filter. But if the original filter includes a query, the new filter must have the same status (public or private) as the original filter. The reason for this is that only public queries can be used with public filters and only private queries can be used with private filters.

To copy a filter

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Do one of the following:
 - ◆ Right-click a column header on upper pane. Then choose **Filters** from the context menu.
 - ◆ Choose **Filters** ► **Filters** from the component or context menu.

The **Filters** dialog appears.

- 4 Select a filter from the **Filters** list box.
- 5 Click **Save As**. The **Save Filter As** dialog appears.
- 6 Type a new filter name.
- 7 Select or clear the **Public** check box. If the filter includes a query, the status of the new filter must be the same as the status of the original filter.
- 8 Click **OK** to return to the **Filters** dialog.
- 9 Do one of the following:
 - ◆ Click **Select** to apply the new filter to the upper pane.
 - ◆ Click **Close** to exit without applying the new filter.

If you want to create a public filter based on a private filter with an associated private query, a work around exists.

To copy a private filter and change its status

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Do one of the following:
 - ◆ Right-click a column header on upper pane. Then choose **Filters** from the context menu.
 - ◆ Choose **Filters** ► **Filters** from the component or context menu.

The **Filters** dialog appears.

- 4 From the list box, select a private filter that has a query attached to it. For private filters, the **Public** check box is clear. The **Query** list box shows the query associated with a particular filter.
- 5 Click the **Query** button. The **Query** dialog appears, with the selected query highlighted.
- 6 Select the **Copy** button. The **Copy Query** dialog appears.
- 7 Type a name for the new query, and select the **Public** check box to change the status of the query. Click **OK**. This action re-displays the **Queries** dialog.
- 8 Highlight the new query, and click **Select**. This action displays the **Filters** dialog.

- 9 Select the filter that has the new public query, and click **Save As**. Type a name for the new filter, and select the **Public** check box. Click **OK**.

This action creates a new public filter with an attached public query, while the original private filter with its private query remains the same.

Related Concepts

[Filters](#)

Related Procedures

[Creating Filters](#)

Related Reference

[Filters Reference Topics](#)

Creating Filters

To limit the data shown on the upper pane, you can create a filter. Filters can be based on the current arrangement of data in the upper pane or created from scratch. Once a filter has been created, it can be used in any project in the same server configuration by any user with the appropriate access rights.

You can also create a new filter based on an existing filter by copying the existing filter.

Note: If you set up a filter and do a [Send to](#) in the client, only the fields displayed by the filter are sent to the recipient.

To save the current arrangement as a filter

- 1 Select a folder from the **Folder Tree**.
- 2 Click a component tab in the upper pane.
- 3 Sort and group the data shown on the upper pane, as desired.
- 4 Right-click a column header in the upper pane and choose [Save Current Settings](#) from the context menu. The **Save Current Settings** dialog box opens.
- 5 Type a name for this filter in the **Filter Name** text box.
- 6 Do one of the following:
 - ◆ Check **Public** to add this filter to the project, so anyone with the appropriate access rights can use it.
 - ◆ Uncheck **Public** to make the filter private, available only to your user ID.
- 7 Click **OK**.

To create a new filter from scratch

- 1 Right-click a column header on upper pane and choose [Filters](#) to open the **Filters** dialog box.
- 2 Click **New** to open the **New Filter** dialog box.
- 3 Type a name for the filter in the **Filter Name** text box.
- 4 Check **Public** if you want to add this filter to the project so that anyone with the appropriate access rights can use it instead of making it available to your user ID.
- 5 Then click **OK**.
- 6 Click any of the following buttons in the **Filters** dialog box and specify the options:
 - ◆ **Fields**: Select the column header fields.
 - ◆ **Sort, Group**: Sort and group items in up to four fields in ascending or descending order.
 - ◆ **Query**: Limit the items that appear in the upper pane to those that match the query.
- 7 Click **Context**, for files only, and specify the files that will be affected by the filter.
Clicking this button opens the **Set Filter Type** dialog box where you apply the filter to one of the following by selecting an option button:
 - ◆ **Items in the view** is equivalent to applying both your filter and the **Files in view** filter.
 - ◆ **Items not in the view** is equivalent to applying both your filter and the **Files not in view** filter.

- ◆ **All items not excluded from the view** is equivalent to applying both your filter and the **<All Non-Excluded Files>** filter.

- 8 Click **OK** to return to the **Filters** dialog box.
- 9 Click **OK** to apply the filter.

Note: If this is a public filter, you can set individual or component-level access rights for it.

Related Concepts

[Filters](#)

Related Procedures

[Applying Predefined Filters](#)

[Copying Filters](#)

[Component-level Filter Access Rights](#)

[Individual Filter Access Rights](#)

[Filtering Items in a View Compare/Merge Session](#)

Related Reference

[Filters Reference Topics](#)

Customizing Change Request Filters

After you have sorted, grouped, selected columns, applied queries to the change requests component in the upper pane, you can save the arrangement of change request data that appears in the upper pane as a filter. You can later apply the filter to any change request data to view the data using the same arrangement.

To create a filter for the current change request arrangement

- 1 Right-click the column headers and select **Save Current Settings** from the context menu. The **Save Current Settings** dialog appears.
- 2 Type a filter name in the **Filter Name** text box.
- 3 Select or clear the **Public** check box depending on whether this filter is to be used by all or only on your workstation.
- 4 Click **OK**. The filter name will appear from now on in the **Filter** drop-down list box.

Related Concepts

[Filters](#)

[Change Requests](#)

Related Procedures

[Creating Filters](#)

[Applying Predefined Filters](#)

[Working with Change Requests](#)

[Customizing Change Request Reports](#)

Related Reference

[Change Request Filters](#)

Deleting Filters

If desired, you can delete filters that you no longer use.

To delete a filter

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Do one of the following:
 - ◆ Right-click a column header on upper pane. Then choose **Filters** from the context menu.
 - ◆ Choose **Filters** ► **Filters** from the component or context menu.

The **Filters** dialog appears.

- 4 Select the filter to be deleted from the **Filters** list box.
- 5 Click **Delete**.
- 6 When a message box asks you to confirm your deletion, click **OK**. This action returns you to the **Filter** dialog.
- 7 Click **Close**.

Related Concepts

[Filters](#)

Related Procedures

[Creating Filters](#)

Related Reference

[Filters Reference Topics](#)

Editing Filters

You edit filters by changing their fields, sort orders, or queries.

To edit a filter

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Do one of the following:
 - ◆ Right-click a column header on upper pane. Then choose **Filters** from the context menu.
 - ◆ Choose **Filters** ► **Filters** from the component or context menu.

The **Filters** dialog appears.

- 4 Select a filter from the **Filters** list box.
- 5 Edit any of the following:
 - ◆ **Fields** button, to select the column header fields.
 - ◆ **Sort, Group** button, to sort and group items in up to four fields in ascending or descending order.
 - ◆ **Query** button, to limit the items that appear in the upper pane to those that match the query.
 - ◆ **Context** button (for files only), to specify the files that will be affected by the filter. Clicking this button opens the Set Filter Type dialog. On this dialog, apply the filter to one of the following by selecting an option button: **Items in the view** is equivalent to applying both your filter and the **Files in view** filter. **Items not in the view** is equivalent to applying both your filter and the **Files not in view** filter. **All items not excluded from the view** is equivalent to applying both your filter and the filter.
- 6 Click **Save As**. The **Save Filter As** dialog box appears. Do *not* change the name of the filter.
- 7 Click **OK** to return to the **Filters** dialog box.
- 8 Do one of the following:
 - ◆ Click **Select** to apply the edited filter to the upper pane.
 - ◆ Click **Close** to exit without applying the edited filter.

Related Concepts

[Filters](#)

Related Procedures

[Creating Filters](#)

Related Reference

[Filters Reference Topics](#)

Filtering Process Tasks From Other Tasks

If you have enabled enhanced process links in your project so that StarTeam is creating process tasks, you can filter your tasks to separate the process tasks from the regular tasks. Use the **Usage** field value to determine the difference between process tasks and standard tasks. If the **Usage** value is anything besides **Other**, then it is a process task.

To filter process tasks from other tasks

- 1 Choose **Task** ► **Filters** ► **Filters**.
This displays the **Filters** dialog box.
- 2 Click **New** and give the new filter a name.
Alternatively, copy an existing filter by selecting it, clicking **Save As**, and giving it a new name. Then select the copied filter and continue with the next steps.
- 3 Click **Fields** in the **Filters** dialog box.
- 4 Move the **Usage** field from the **Available Fields** list to the **Show Fields in this Order** list and click **OK**.
- 5 Click **Query** to open the **Queries** dialog box, and click **New**.
This opens the **Edit Query** dialog box.
- 6 Type a **Name** for the new query, and choose the following in the **Condition Node** section: **Field = Usage**, **Operator = Not Equal**, and **Value = Other**.
- 7 Click **Add** to add the condition to the query, and **Save** to save the query and return to the **Queries** dialog box.
Your new query is now highlighted in the list of queries.
- 8 Click **Select** in the **Queries** dialog box use this query in your new filter.
You are returned to the **Filters** dialog box, and your new filter should be highlighted.
- 9 Click **Save As** to save the filter.
Notice your new filter has been added to the **Filter** drop-down list at the top of the client.

To use the filter, simply select it from the **Filter** list on the **Task** tab.

Note: Conversely, you can create a filter that displays only the standard tasks. In the query, use the condition **Usage Equals Other**.

Tip: If you use tasks on a regular basis, and not just for process tasks, add **Usage Equals Other** to existing queries so you never see process tasks when working on tasks that have been manually created, or imported from Microsoft Project using Borland's Microsoft Project integration.

Related Concepts

[Process Items and Process Links](#)
[Process Tasks and Enhanced Process Links](#)

Related Procedures

[Creating Filters](#)
[Creating Queries](#)

Resetting Filters

In StarTeam, you can apply a filter, then rearrange the data on the upper pane or apply a new query. Doing this places an asterisk in front of the filter's name, showing that it has been changed. After looking at the new data, you can then reset the filter as it was originally defined on the server, which removes the asterisk.

To restore (reset) a filter to its original definition

- 1 Do one of the following when the filter in the list box has changed:
 - ◆ Right-click a column header on the upper pane, then select **Reset Current Settings** from the context menu.
 - ◆ Choose **Filters** ► **Reset Current Settings** from the component or context menu.

The system asks: `Reset filter: Filter>?`

- 2 Click **OK**. This action resets the filter and removes the asterisk.

Related Concepts

[Filters](#)

Related Procedures

[Creating Filters](#)

Related Reference

[Filters Reference Topics](#)

Showing Fields in a Change Request

You can select which fields are displayed for a change request.

To show specific fields in a change request

- 1 Right-click the column headers and select **Show Fields** from the context menu.
This opens the **Show Fields** dialog box.
- 2 Make sure the **CR Number**, **Entered By**, and any other appropriate fields are displayed in the **Show these fields in this order** list.
- 3 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Sorting and Grouping Change Requests

You can sort change requests by the data in a particular column or group the change requests.

To sort or group change requests

- 1 Click on a column header on the Change Requests tab to sort the change requests by the data in that column.

Note: If you want to sort or group the change requests first by the data in one column and then by the data in another column, see the following steps. You can sort or group the change requests in up to four levels of groupings.

- 2 To sort or group the change requests in multiple levels, right-click the column headers and select **Sort and Group** from the context menu. The **Sort and Group** dialog box opens.
- 3 From the **First By** drop-down list box, select a column title.
- 4 Optionally, group the change requests by the data in this column, select the **Group By** check box.

If you select the **Group By** check box, the change requests are grouped together in nested lists and you must drill down to view the change requests in each group. If you do not select the **Group By** check box, the change requests are all displayed on the **Change Requests** tab, sorted by your choices in **Sort and Group** dialog box.

Note: By default, the column data is sorted or grouped based on the internal key or order. You can use the **Sort Options** button and choose to sort or group the data based on the text and optionally, case sensitivity.

- 5 Repeat steps three and four to define up to four levels of sort orders or groupings.
- 6 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Sorting and Grouping Data

You can choose to do a primary sort in the upper pane (based on one column), or a more complicated sorts up to a fourth order.

To do a primary sort on one column

- 1 Open the view on the data you wish to sort or group.
- 2 Click a column header to sort the data in the upper pane based on the value in that column.
The sort is in ascending order by number, letter, internal order, or internal key, depending on the data.
- 3 Click the column header again to reverse the sort order.

A triangle appears on column header of the sorted column. The triangle points upward for ascending sorts and downward for descending sorts.

Note: You can also sort the data in the lower pane when the **Link** tab is selected.

To perform up to a fourth-order sort

- 1 Do one of the following:
 - ◆ Right-click a column header on upper pane and choose **Sort and Group** from the context menu.
 - ◆ Right-click in the upper pane and choose **Filters** ► **Sort and Group** from the context menu.

The **Sort and Group** dialog box displays four group boxes, each indented slightly more to the right than the one above it. The first group box designates a primary sort order, the second designates a secondary sort, and so on.

- 2 Optionally, check the **Show Advanced Fields** check box at the bottom of the dialog box to list all the fields in **First By** and **Then By** drop-down list boxes. Some fields are rarely used and considered advanced.
- 3 Select a field from the **First By** drop-down list box.
If you are grouping the items, the field does not need to be displayed in the upper pane. If you are not grouping the items, you can sort them based on a field that is not displayed, but you will not be able to tell where one group leaves off and the next begins.
- 4 Select the **Ascending** or **Descending** option button. The default setting is ascending order.
- 5 Select **Group By** to group the items which have the same values in this field.
If you do not select any additional sort options, text fields are sorted in ASCII order. Enumerated and user ID fields are sorted by their internal order or internal keys. That is, enumerated fields are sorted in the order given to them by the person who created the field; user ID fields are sorted in the order in which they were created. The application disables the Sort Options button for numeric and date/time fields.
- 6 Optionally, click **Sort Options** for additional sorting selections. The **Sort Options** dialog box appears.
 - ◆ Select **As Text** to sort enumerated and user ID fields by the names of their possible values. For text fields, **As Text** is your only choice.
 - ◆ Uncheck the **Case-sensitive** check box to sort alphabetically or check it to sort in ASCII order (where uppercase letters precede lowercase letters).
- 7 Add secondary and lower order sorts by using the **Then By Group** boxes as needed.

Related Concepts

[Cross-Platform Client Overview](#)

[Displaying Additional Fields](#)

Related Procedures

[Filtering Data](#)

[Creating Filters](#)

[Creating Queries](#)

[Creating Reports](#)

Specifying Process Rules and Process Items

This section contains tasks related to process rules and process items.

In This Section

[Displaying Only Enhanced Process Links](#)

Describes how to display enhanced process links in the Link pane.

[Establishing Process Rules for Projects](#)

Describes how to require that files added to or checked in to a project be linked to a process item.

[Filtering Process Tasks From Other Tasks](#)

Describes how to filter process tasks from other tasks.

[Finding Files Linked to Active Process Items](#)

Describes how to find all the files linked to an active process item.

[Linking Files to Process Items](#)

Describes the process of linking and pinning files to process items and to active process items..

[Promoting File Changes Into Baselines](#)

Describes how to promote file changes into baselines.

[Viewing Links](#)

Describes how to view links in the Link pane.

[Viewing Process Rules](#)

Describes how to review the current project settings for process rules.

Displaying Only Enhanced Process Links

If your project has enhanced process links enabled, you can choose to display only enhanced process links in the **Link** pane.

To display only enhanced process links in the Link pane

- 1 Click the **Link** tab at the bottom of the client.
- 2 Select the radio button **Show enhanced links only** at the top of the **Link** pane.

Now when you select an item in the upper pane that has links, the **Link** pane only shows enhanced links, not standard links.

Related Concepts

[Overview of Projects](#)

[Process Items and Process Links](#)

Related Procedures

[Enabling the use of Enhanced Process Links](#)

Establishing Process Rules for Projects

Establishing a system of process rules allows you to:

- ◆ Require that process items are used every time files are added or checked into the project.
- ◆ Stipulate that only certain types of items with specific statuses can be used as process items in the project.
- ◆ Enable the use of enhanced process links for the project.

Note: To set process rules, you must have the access rights required to change project properties. Usually, only team leaders and administrators have these rights. You must also verify that project users have the rights to see and modify items in the project view, to create and modify links on files and process items, and to create tasks and link to tasks if using the enhanced model..

To require use of process items at check-in, and specify which types to use

- 1 Choose **Project** ► **Properties** and click the **Process Rules** tab in the **Project Properties** dialog box.
- 2 Check **Require Selection Of Process Items When Files Are Added Or Checked In**.
- 3 Check which types of items you want to allow for use as process items. (See below.)

To permit the use of change requests as process items

- 1 Check **Permit Selection Of Change Requests As Process Items**.
- 2 Do one of the following:
 - ◆ Check **Open** to use only change requests with **Open** status.
 - ◆ Check **In Progress** to use only change requests with **In Progress** status.
 - ◆ Check both **Open** and **In Progress** to use only change requests with either of these statuses.
 - ◆ Uncheck both **Open** and **In Progress** to allow any change request to be used as a process item, regardless of status.

To permit the use of requirements as process items

- 1 Check **Permit Selection Of Requirements As Process Items**.
- 2 Do one of the following:
 - ◆ Check **Approved** to use only requirements with **Approved** status.
 - ◆ Uncheck **Approved** to allow any requirement to be used as a process item, regardless of status.

To permit the use of tasks as process items

- 1 Check **Permit Selection Of Tasks As Process Items**.
- 2 Do one of the following:
 - ◆ Check **Ready To Start** to use only tasks with **Ready To Start** status.

- ◆ Check **In Progress** to use only tasks with **In Progress** status.
- ◆ Check both **Ready To Start** and **In Progress** to use only tasks with either of these statuses.
- ◆ Uncheck both **Ready To Start** and **In Progress** to allow any tasks to be used as a process item, regardless of status.

To enable enhanced process links for the project

- 1 Choose **Project** ► **Properties** and click the **Process Rules** tab in the **Project Properties** dialog box.
- 2 Check or uncheck **Enable Enhanced Process Links** to specify whether to use the standard linking model or the enhanced linking model.

In the enhanced model, the process item (that is, the item specified as the reason for making a given set of changes) is linked to a process the task, which is automatically created to represent the associated changes in a particular view. Changes are linked to the process item *indirectly*, through a process task.

Note: Some StarTeam integrations do not recognize process rules and will ignore them.

Related Concepts

[Process Items Overview](#)

[Process Items and Process Links](#)

Related Procedures

[Assigning Access Rights to Projects](#)

Related Reference

[Access Rights and Privileges](#)

Filtering Process Tasks From Other Tasks

If you have enabled enhanced process links in your project so that StarTeam is creating process tasks, you can filter your tasks to separate the process tasks from the regular tasks. Use the **Usage** field value to determine the difference between process tasks and standard tasks. If the **Usage** value is anything besides **Other**, then it is a process task.

To filter process tasks from other tasks

- 1 Choose **Task** ► **Filters** ► **Filters**.
This displays the **Filters** dialog box.
- 2 Click **New** and give the new filter a name.
Alternatively, copy an existing filter by selecting it, clicking **Save As**, and giving it a new name. Then select the copied filter and continue with the next steps.
- 3 Click **Fields** in the **Filters** dialog box.
- 4 Move the **Usage** field from the **Available Fields** list to the **Show Fields in this Order** list and click **OK**.
- 5 Click **Query** to open the **Queries** dialog box, and click **New**.
This opens the **Edit Query** dialog box.
- 6 Type a **Name** for the new query, and choose the following in the **Condition Node** section: **Field = Usage**, **Operator = Not Equal**, and **Value = Other**.
- 7 Click **Add** to add the condition to the query, and **Save** to save the query and return to the **Queries** dialog box.
Your new query is now highlighted in the list of queries.
- 8 Click **Select** in the **Queries** dialog box use this query in your new filter.
You are returned to the **Filters** dialog box, and your new filter should be highlighted.
- 9 Click **Save As** to save the filter.
Notice your new filter has been added to the **Filter** drop-down list at the top of the client.

To use the filter, simply select it from the **Filter** list on the **Task** tab.

Note: Conversely, you can create a filter that displays only the standard tasks. In the query, use the condition **Usage Equals Other**.

Tip: If you use tasks on a regular basis, and not just for process tasks, add **Usage Equals Other** to existing queries so you never see process tasks when working on tasks that have been manually created, or imported from Microsoft Project using Borland's Microsoft Project integration.

Related Concepts

[Process Items and Process Links](#)
[Process Tasks and Enhanced Process Links](#)

Related Procedures

[Creating Filters](#)
[Creating Queries](#)

Finding Files Linked to Active Process Items

When you have files linked to an active process item, you can quickly find all the files linked to it.

To find files linked to an active process item

- 1 Click the **Change Request**, **Requirement**, or **Task** tab in the upper pane that contains the active process item.
You can see what item is the active process item by looking at the left side of the **Status Bar**. The second box in the **Status Bar** displays the **Active Process Item** icon, followed by the name of the item.
- 2 Select the active process item and click the **Link** tab in the lower pane.

The **Link** pane displays all the files linked to that item.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)

Linking Files to Process Items

If process rules are enforced for a project, linking and pinning new file revisions to a process item is required. Otherwise, this step is optional, and you can select any change request, requirement, or task as a process item.

To link and pin a file revision to a process item

- 1 Click the **File** tab in the upper pane and select a file or files to be checked in.
- 2 Choose **File** ► **Check In** to display the **Check In** dialog box.
- 3 Check **Link and Pin Process Item**.
- 4 Click **Select** to open the **Select Process Item** dialog box.
- 5 Select one of the following to limit the list of possible process items:
 - ◆ **List All Permitted Items**: Displays all items that can be used as process items. If process rules are not enforced, the list contains all change requests, requirements, and tasks.
 - ◆ **List All Permitted Items Assigned To Me**: Displays all the items for which you are responsible that can be used as process items.
 - ◆ **List Linked Items**: Lists the process items that are already linked to at least one of the files you are checking in. No process item appears on the list more than once, even if it is linked to several files. Also, when a process item is linked to more than one file, the dialog box displays the name of only one file. Despite this fact, the application will update or create links for every file being added.
- 6 Select the **Change Request**, **Requirement**, or **Task** tab to restrict the list to a specific type of item.
- 7 Select one item in the list to be the active process item and click **OK**.
- 8 Optionally, in the **Check In** dialog box, check **Mark Selected Item As Fixed/Finished/Complete** if work on the process item is completed.
- 9 Complete filling in the fields in the **Check In** dialog box and press **OK**.

Tip: If process rules are enforced, the use of some change requests, requirements, or tasks as process items may not be permitted because of their status. If you select such an item and click **OK**, the application notifies you of this fact. By double-clicking the item in the list box, you can display its properties and change it to a permitted status for a linked process item.

To link and pin a file revision to the active process item

- 1 Click the **File** tab in the upper pane and select a file or files to be checked in.
- 2 Choose **File** ► **Check In** to display the **Check In** dialog box.
If you have already specified a process item, it will appear by default in the **Link and Pin Process Item** field.
- 3 Check **Link and Pin Process Item** to accept this active process item.
- 4 Optionally, in the **Check In** dialog box, check **Mark Selected Item As Fixed/Finished/Complete** if work on the active process item is completed.
- 5 Complete filling in the fields in the **Check In** dialog box and press **OK**.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Customizing Link Properties](#)
[Deleting Links](#)

Promoting File Changes Into Baselines

Process rules are useful when creating baseline builds or configurations. A build is a labeled configuration that identifies the file revisions and process items that define the code and content baseline.

Process rules require that each new file revision be linked to a process item, which allows the development team to promote these changes into baselines.

If process rules are not enforced, developers using the application can create baselines in two ways:

- ◆ By labeling an entire project view at a specific point in time.
- ◆ By associating file revisions with a revision label on check-in.

To promote file changes into baselines

- 1 Start with the previous baseline (for example, check it out based on its label.)
- 2 Select process items for inclusion in the new baseline.
- 3 Label the new baseline.

Even if process rules are not enforced, individual users can use process items for tracking purposes when adding or checking in files.

Related Concepts

[Process Items Overview](#)
[Change Requests](#)
[Requirements](#)
[Tasks](#)

Related Procedures

[Working with Change Requests](#)
[Linking Items Internally or Externally](#)
[Reviewing Linked Change Requests](#)
[Setting Active Process Items](#)
[Finding Files Linked to Active Process Items](#)
[Establishing Process Rules for Projects](#)
[Viewing Process Rules](#)
[Assigning Access Rights to Projects](#)

Related Reference

[Access Rights and Privileges](#)

Viewing Links

StarTeam displays links at the bottom of the client in the **Link** pane for any item selected item in upper pane.

To view links

- 1 Select an item in the upper pane.
- 2 Click the **Link** tab at the bottom of the client.

You can control whether to display all links, or only enhanced links by selecting the appropriate option at the top of the **Link** pane.

You can also use the drop-down list box in the **Link** pane (called **Linked Item**) that identifies the item for which links are displayed. Initially, the item displayed is the one selected in the upper pane. To select a different linked item, click the arrow beside the **Linked Item** list box

Note: You can drag any link displayed in the Link tab to the **Linked Item** list box. This action changes the linked item as well as the links displayed in the **Link** pane. This feature is especially helpful in viewing the links to process tasks. If you switch to a different tab and then return, the list box reverts to displaying only the item selected from the upper pane.

When you select a process item in the upper pane, the only link displayed in the **Link** pane may be the link to a process task. To see the items that are linked to the process task (and are therefore linked indirectly to your process item), drag the process task link from the list in the **Link** pane onto the **Linked Item** list box. The **Linked Item** list box now displays the process task, and the **Link** pane list now displays the process task links. For example, the **Link** pane could show a link to each file checked in using the previously selected process item, as well as a link to the process item itself.

Related Concepts

[Overview of Projects](#)

[Process Items and Process Links](#)

Related Procedures

[Enabling the use of Enhanced Process Links](#)

Viewing Process Rules

If process rules are not enforced, any change request, requirement, or task can be used as a process item, regardless of its status. However, if process rules are enforced, you may be able to select only one type of item as a process item. In addition, acceptable process items may be limited to those with specific statuses.

You can determine whether process rules are in effect for a specific project and what those rules are by reviewing project properties. If you do not have the access rights necessary to do this, ask your administrator what process items apply to the project and what restrictions have been placed on them.

To review project properties for process rules

- 1 Choose **Project** ► **Properties** to open the **Project Properties** dialog box.
- 2 Click the **Process Rules** tab.
- 3 Review the restrictions.
If no statuses have been selected for a permitted process item, then any status can be used.
- 4 Click **OK** to close the **Project Properties** dialog box.

Related Concepts

[Process Items Overview](#)
[Change Requests](#)
[Requirements](#)
[Tasks](#)

Related Procedures

[Working with Change Requests](#)
[Linking Items Internally or Externally](#)
[Reviewing Linked Change Requests](#)
[Setting Active Process Items](#)
[Finding Files Linked to Active Process Items](#)
[Establishing Process Rules for Projects](#)
[Promoting File Changes Into Baselines](#)
[Assigning Access Rights to Projects](#)

Related Reference

[Access Rights and Privileges](#)

Accessing Projects and Items with Shortcuts and URLs

This section contains tasks related to using shortcuts and URLs to access projects and items.

In This Section

[Copying and Opening URLs](#)

Describes how to copy and open a URL to a StarTeam project, view, folder, or item.

[Creating Shortcuts](#)

Describes how to save a shortcut to an item.

Copying and Opening URLs

The StarTeam client can open URL shortcut links to projects, views, folders, Not-in-View folders, and items. This enables you to quickly access specific locations in a project. You can also copy a URL to the clipboard, which allows you to easily move its contents to an appropriate application, such as an email client or a document.

Like other URLs, StarTeam URLs include the name of the server for the connection. In some organizations, StarTeam servers may be reached from both the Internet and the corporate intranet. In such cases, a server may have two different IP addresses. If you configure the server list to reference a server by its IP address, rather than its DNS name, then any URLs generated by the client will work only from the network on which that IP address exists.

URL displays can be changed in your **Personal Options** settings.

Note: StarTeam shortcuts use the file extension `.stx`. For shortcuts to saved View Compare/Merge sessions, it uses `.vcmx` and `.vcms`.

To copy a URL to the clipboard

- 1 Select one or more items in the upper pane.
- 2 Right-click the selected items and choose **Copy URL to clipboard**.

This action places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.

- 3 Paste the URL to the application of choice.

Note: Copying a URL to the clipboard is equivalent to dragging an item or items from the list pane or folder tree onto an application. Not all applications support pasting the HTML representation, although Word, Excel, and Outlook do support HTML data.

To open a URL

- 1 Choose **Project** ► **Open StarTeam URL**.
- 2 Type a valid URL to a project, view, folder, or item in the **Open StarTeam URL** dialog box.

For example: `starteam:/hostname:49201/myproject`.

- ◆ If the URL is a reference to a project, the default view of the project opens.
- ◆ If the URL is a reference to a view or folder, then the view or folder opens.
- ◆ If the URL is a reference to an item, the item's view opens, the item's parent folder is selected in the folder tree, the item type is selected, and the item itself is selected in the item list or tree on the upper pane.

Tip: Double-click a StarTeam URL link in an email or other application to open StarTeam to the specified project, view, or item. In Windows Explorer, StarTeam links display a small StarTeam icon which you can double-click to open.

Related Procedures

[Opening a Saved or Exported View Compare/Merge Session](#)

Creating Shortcuts

For easy access to items that you are tracking, you can save shortcuts to them on your desktop. Opening a shortcut starts the application, opens the project view in the configuration it had when the shortcut was created, and displays the **Properties** dialog box for the item.

To save a shortcut to an item

- 1 Select a folder from the **Folder Tree**.
- 2 Click the appropriate component tab.
- 3 Right-click an item in the upper pane and choose **Save Shortcut**.
This opens the **Save As** dialog box.
- 4 Type a name for the shortcut in the **File Name** text box if you do not want to use the default name.
Be sure to keep the `.stx` extension.
- 5 Select a location, usually your desktop, for storing the shortcut and click Save.

Note: To open the item from the shortcut you saved, double-click the shortcut file (`.stx`).

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Files to Projects](#)

[Saving Projects as Shortcuts](#)

[Opening Projects with Shortcuts](#)

Working with Revisions

This section contains tasks related to working with revisions.

In This Section

[Checking Out Historical Versions of Files](#)

Describes how to check out a previous revision of a file.

[Comparing Historical File Contents](#)

Describes how to compare the contents of two historical files in the repository.

[Comparing Historical Properties](#)

Describes how to compare the properties of two historical item revisions.

[Comparing Properties](#)

Describes how to compare the properties of the tip revision with a previous revision, or two historical revisions.

[Copying Revision Labels](#)

Describes how to copy an existing revision label.

[Creating Revision Labels](#)

Describes how to create a new revision label.

[Detaching a Label from a Rolled-back View](#)

Describes how to detach a label from a view that has been rolled-back to a previous revision.

[Detaching a Label from a Specific Revision](#)

Describes how to detach a label from a specific revision.

[Displaying Historical Properties](#)

Describes how to display the properties of a historical revision.

[Editing Check-in Comments](#)

Describes how to edit check-in comments after the check-in has been done.

[Linking Specific Revisions](#)

Describes how to link to specific revisions.

[Viewing Previous File Revisions](#)

Describes how to view a previous file revision.

Checking Out Historical Versions of Files

You can easily check out a previous revision of a file using the **History** pane. You will have a choice whether to check out the file to the current working folder, which would overwrite the current file, or to check out to a different location.

To check out a previous revision of a file

- 1 Click the **File** tab in the upper pane and select the file.
- 2 Click the **History** tab or the **Label** tab in the lower pane, and select the revision to check out.
- 3 Right-click the selected file and choose **Check Out** or **Check Out To**.
This opens the **Check Out** dialog box.
- 4 Use the **Check Out** dialog box to check out the file as described in the procedure above for checking out files.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Comparing Historical File Contents](#)
[Comparing Historical Properties](#)
[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Comparing Historical File Contents

You can only compare historical file contents using the main or embedded File Compare/Merge in the StarTeam client. You cannot edit historical file revisions.

To compare the contents of two historical revisions in the repository

- 1 Click the **History** tab and select two revisions to compare.
- 2 Choose **Tools** ► **Compare**.

The embedded compare panes display the contents of the two files at the bottom of the StarTeam window.

Note: You can also compare the properties of two non-file items, such as a change request, or two revisions of the same non-file item. Select the two non-file items, then choose **Tools** ► **Compare**. This opens the embedded double-paned compare view at the bottom of the window and displays the properties of the two selected items.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Two Local Files](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)

Comparing Historical Properties

You can compare the properties of two historical revisions of an item using the **History** pane.

To compare the properties of two historical file revisions

- 1 Select the item in the upper pane whose historical properties you want to compare.
You can select an item on any of the component tabs in the upper pane except the **Audit** tab.
- 2 Click the **History** tab and select the two item revisions.
Use **CTRL+CLICK** to select the second revision.
- 3 Right-click the selection and choose **Compare ▸ Properties**.

The **Compare Properties** window displays the properties of the revisions in adjacent columns.

Note: You can also select two non-file items, such as a change request, or two revisions of the same non-file item, and choose **Tools ▸ Compare**. This opens the embedded double-paned compare view at the bottom of the window and displays the properties of the two selected items.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Properties](#)

[Comparing Historical File Contents](#)

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Comparing Properties

Using the **History** pane, you can compare item properties of the tip revision with a previous revision, or two historical revisions.

To compare the item properties of the tip revision with a previous revision

- 1 Select the item in the upper pane whose properties you want to compare.
You can select an item on any of the component tabs in the upper pane except the **Audit** tab.
- 2 Click the **History** tab in the lower pane and select any revision except the latest (tip) revision.
- 3 Right-click the selected revision and choose **Compare** ► **Properties**.

The **Compare Properties** window displays the properties of the tip revision and the selected historical revision in adjacent columns.

To compare the properties of two historical file revisions

- 1 Select the item in the upper pane whose historical properties you want to compare.
You can select an item on any of the component tabs in the upper pane except the **Audit** tab.
- 2 Click the **History** tab in the lower pane and select the two historical revisions whose properties you want to compare.
Use **CTRL+CLICK** to select the second revision.
- 3 Right-click the selection and choose **Compare** ► **Properties**.

The **Compare Properties** window displays the properties of the selected historical revision in adjacent columns.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Properties](#)
[Comparing Historical File Contents](#)
[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Copying Revision Labels

Occasionally, you may want to copy a revision label. For example, if you move or share an item from one view (source view) to another (target view), labels from the source view do not become part of target view. However, by copying the revision labels after the move or share, you can selectively maintain revision labels on the moved or shared items.

Copying a revision label immediately attaches it to the same revisions of the same items as the original revision label. If the two revision labels are in the same view, each label will be attached to the same number of items. However, if the two revision labels are in different views, the new label becomes attached to the same revisions of the same items only if the items and their revisions exist in the new label's view at the time of the copy operation.

Although you can copy revision labels in a variety of ways, the following procedure allows you to copy a revision label whether it is in the current view or in another accessible view. It assumes that you are dealing with files, but can be adapted for other types of items.

To create a revision label based on an existing label (with a few additions or exceptions)

- 1 Choose **View** ► **Labels** and select the **Revision** tab in the **Labels** dialog box.
- 2 Click **New** to open the **Revision Label** dialog box.
- 3 Type a name and description for the revision label.
- 4 Check **Copy From Another Revision Label**.
- 5 Click **Select** to open the **Copy a Revision** dialog box.
- 6 Select a project from **Project** from the list, a view in the **View** tree, and a revision label in the **Labels** list.
- 7 Click **OK** to close the **Copy a Revision Label** dialog box, and click **OK** to close the **Revision Label** dialog box. Click **Close** on the **Labels** dialog box.

The new revision label is attached to the same revisions as the existing label.

- 8 Do one of the following:
 - ◆ Check in the changed file or files using the new revision label.
 - ◆ Check in the changed file and attach the new revision label manually to the changed file revisions. To do this, select the checked in file and click the **Label** tab in the lower pane. Drag the new revision label to the correct (probably tip) revision in the **Label** pane. Repeat for any other changed files.

Note: If you have added a new file, use **File** ► **Labels** ► **Attach** to attach the label.

Related Concepts

[Labels](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Attaching Labels to Folders](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Creating Revision Labels

Like view labels, new revision labels can be created from the **View** menu. In fact, if you are creating a new revision label based on an existing revision label in another view, you must use the **View** menu for this purpose. However, revision labels can also be created from the **Folder Tree** menu, a component menu, or the context menu.

To create a new revision label for selected items (from the View menu)

- 1 Select a folder in the folder hierarchy tree.
- 2 Select one or more items on any of the tabs in the upper pane.
- 3 Choose **View** ▸ **Labels** and click the **Revision** tab in the **Labels** dialog box.
The labels are listed in reverse chronological order based on the time at which they were created.
- 4 Type a name and description for the label.
The maximum label name length is 64 characters and the description length is 254 characters.
- 5 Optionally, check **Frozen** to freeze the label so that revisions attached to it cannot be changed.
- 6 Click **OK**.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)
[Attaching Labels to Folders](#)
[Creating View Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Deleting Labels](#)
[Detaching Labels from Items](#)

Detaching a Label from a Rolled-back View

Sometimes you need to detach a label from an item in a rolled-back view. For example, suppose you deleted a file that had view labels attached to it. Later you created a build label based on one of the view labels that was attached to the deleted file. If you roll back the view to the new build label in order to perform a build, the deleted file reappears in your view. If you do not want that file in this build, you can detach the new build label from that file. If you try to detach any other label from the rolled-back view, an error message informs you that you can detach only the label to which the view has been rolled back.

Note: You can attach and detach any labels from items in current view configurations, but you cannot see deleted items in those configurations. You can detach view labels from deleted items only if you roll back the view to a configuration based on the label you want to detach.

To detach a view label from a in a rolled-back view

- 1 Roll back the view to the view label you want to detach.
- 2 From the upper pane, select the item from which the view label is to be detached.
- 3 Click the **Label** tab in the lower pane.
- 4 Right-click the label you wish to remove and choose **Detach**.
- 5 Click **OK**.

The item from which the label is detached will disappear after a refresh.

Related Concepts

[Labels](#)
[Folders](#)
[Overview of Views](#)
[Understanding Branching](#)
[Proper Use of Views](#)

Related Procedures

[Detaching Labels from Folders](#)
[Detaching Labels from Items](#)
[Attaching Labels to Items](#)
[Creating View Labels](#)
[Creating Revision Labels](#)
[Copying Revision Labels](#)
[Reviewing and Moving Labels](#)
[Freezing or Unfreezing Labels](#)
[Promoting View Labels](#)
[Demoting View Labels](#)
[Configuring or Viewing Label Properties](#)
[Deleting Labels](#)

Detaching a Label from a Specific Revision

If you decide not to include certain items in a view or revision label, you can detach the label from those items individually or as a group. Generally, the items from which labels are detached are files or folders.

To detach a label from a specific item revision

- 1 Select a folder from the folder hierarchy.
- 2 Click the component tab in the upper pane containing the item you want to detach, and select the item.
- 3 Click the **Label** tab in the lower pane.
- 4 Double-click a revision in the **Label** pane to see all labels attached to it as children of the revision.
- 5 Right-click the label you wish to remove and choose **Detach**.
- 6 Click **OK**.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Copying Revision Labels](#)

[Reviewing and Moving Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

Displaying Historical Properties

You can display the properties for a previous item revision using the **History** pane.

To compare the item properties of a historical revisions

- 1 Select the item in the upper pane whose historical properties you want to view.
You can select an item on any of the component tabs in the upper pane except the **Audit** tab.
- 2 Click the **History** tab in the lower pane.
- 3 Right-click the historical revision in the lower pane and choose **Properties**.

The item's **Properties** dialog box displays the properties of that revision.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Properties](#)

[Comparing Historical File Contents](#)

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Editing Check-in Comments

Normally you enter comments during the check-in process describing the changes you are checking in. You can also add or edit check-in comments after files have been checked in.

To add or edit a check-comment after a file has been checked in

- 1 Select the file in the upper pane on the **File** tab, and click the **History** tab or the **Label** tab.
- 2 Select the revision for which you want to add or edit the comment in the **History** or **Label** pane.
- 3 Right-click the selected revision and choose **Edit Comment** to open the **Edit Comment** dialog box.
- 4 Type the text you want for the comment and click **OK**.

Comment fields allow up to 30,000 characters.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Linking Specific Revisions

Each end of a link has an associated start revision and an end revision that determines the range of revisions to which the link applies. The start revision is always fixed at the time of the creation of the link and is set to the first revision on the current branch. The end revision is under the user's control and may be fixed (or pinned), which puts an upper bound on the linked revisions, or floating, which does not. If a link end is pinned, it is always attached to the same version of the linked folder or file. If a link end floats, it moves from revision to revision, as new revisions of the linked folder or item are created.

By default, a link connects the tip revisions of the linked pair. The revisions selected for both links appear as columns on the **Link** pane.

Determining whether a link is visible on a given item is simple. If any of the revisions between the start and the end revision defined for the link are in the history of the selected item, it is visible. Otherwise, it is not.

To link to a tip revision

- 1 Select a folder or item for which you have created a link.
- 2 Do one of the following:
 - ◆ If you selected an item in the upper pane, click the **Link** tab in the lower pane.
 - ◆ If you selected a folder in the **Folder Tree**, choose **Folder Tree** ► **Properties** to display the **Folder Properties** dialog box, and click the **Link** tab.
- 3 Select one or more links in the **Link** pane.
- 4 Right-click the selected link(s) and choose one of the following options:
 - ◆ **Pin Link** ► **To Source Item At Tip**: This command pins the link to the tip revision of the source (that is, the folder whose properties you are reviewing or the item selected from the upper pane).
 - ◆ **Pin Link** ► **To Target Items At Tip**: This command pins the link to the tip revision of the target (that is, the folder or item in the Item column on the Link pane).
 - ◆ **Pin Link** ► **To Source And Target Items At Tip**: This command pins the link to the tip revisions of both the source and target.
 - ◆ **Float Link** ► **To Source Item**: This command allows the link to float from tip revision to tip revision of the source as new revisions are created.
 - ◆ **Float Link** ► **To Target Items**: This command allows the link to float from tip revision to tip revision of the target as new revisions are created.
 - ◆ **Float Link** ► **To Source And Target Items**: This command allows both the source and the target of the link to float from tip revision to tip revision.

To link to a specific revision

- 1 Right-click a folder or item for which you have created a link.
- 2 Do one of the following.
 - ◆ If you selected an item, click the **Link** tab in the lower pane.
 - ◆ If you selected a folder in the **Folder Tree**, choose **Folder Tree** ► **Properties** to display the **Folder Properties** dialog box, and click the **Link** tab.
- 3 Select a link.

- 4 Right-click the selected link and choose [Link Properties](#).

The **Link Properties** dialog box opens.

- 5 Optionally, type a description or comment about the link in the **Comment** text box.

This text will appear in the **Comment** column of the **Link** pane.

- 6 Do one of the following in the **Source Item** group box:

- ◆ Click **Pin** to display the **Select Version** dialog box. Select a specific folder or item revision from the list. This revision number will appear in the **Selection Version** column of the **Link** pane.
- ◆ Click **Float** so that the link is always connected to the tip revision of this item.

- 7 Do one of the following in the **Is Linked to Target Item** group box:

- ◆ Click **Pin** to display the **Item Version** dialog box. Select a specific folder or item revision from the list. This revision number will appear in the **Item Version** column of the **Link** pane.
- ◆ Click **Float** so that the link is always connected to the tip revision of this item.

- 8 Click **OK**.

Note: You can link items from a project view on one server to an item in another project or view on a different server. This is called an external link.

Related Concepts

[Links: Internal and External](#)

[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)

[Reviewing Linked Change Requests](#)

[Checking Linked Files In and Out](#)

[Selecting Linked Files](#)

[Customizing Link Properties](#)

[Deleting Links](#)

[Linking Files to Process Items](#)

Viewing Previous File Revisions

You can review the contents of a prior file revision in either the default editor or in the application for which the file type is registered.

To review a revision's contents

- 1 On the **File** tab, select the file for which you want to review a revision.
- 2 On the **History** or **Label** tab in the lower pane, select the specific revision you want to review.
- 3 Right-click the selected item to open the context menu and choose one of the following:
 - ◆ **View Revision Content** to copy the revision to a temporary file and display it in the default editor (Notepad or the alternate editor specified in the **Personal Options** dialog box).
 - ◆ **Open Revision Content** to copy the revision to a temporary file and display it in the associated application.

Note: The client creates the temporary files in the local temp directory on the system. For example, if working on a Windows system, the temporary files are created in the `C:\Documents and Settings\<user>\Local Settings\Temp` directory. When you exit the client, the files are deleted from the system.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Working with Properties

This section contains procedures related to working with item properties.

In This Section

[Comparing Historical Properties](#)

Describes how to compare the properties of two historical item revisions.

[Comparing Properties](#)

Describes how to compare the properties of the tip revision with a previous revision, or two historical revisions.

[Configuring or Viewing Label Properties](#)

Describes how to access label properties so you can view or modify them.

[Customizing Link Item Properties](#)

Describes how to modify item properties from a link.

[Customizing Link Properties](#)

Describes how to modify link properties.

[Displaying Historical Properties](#)

Describes how to display the properties of a historical revision.

[Emailing Item Properties](#)

Describes how to email item properties to another team member.

[Using Embedded Property Editors](#)

Describes how to use embedded property editors, which allow you to view, edit, and compare multiple StarTeam items at once.

[Viewing or Modifying Item Properties](#)

Describes how to view or modify properties for items using the standard properties dialog box.

Comparing Historical Properties

You can compare the properties of two historical revisions of an item using the **History** pane.

To compare the properties of two historical file revisions

- 1 Select the item in the upper pane whose historical properties you want to compare.
You can select an item on any of the component tabs in the upper pane except the **Audit** tab.
- 2 Click the **History** tab and select the two item revisions.
Use **CTRL+CLICK** to select the second revision.
- 3 Right-click the selection and choose **Compare** ► **Properties**.

The **Compare Properties** window displays the properties of the revisions in adjacent columns.

Note: You can also select two non-file items, such as a change request, or two revisions of the same non-file item, and choose **Tools** ► **Compare**. This opens the embedded double-paned compare view at the bottom of the window and displays the properties of the two selected items.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Properties](#)

[Comparing Historical File Contents](#)

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Comparing Properties

Using the **History** pane, you can compare item properties of the tip revision with a previous revision, or two historical revisions.

To compare the item properties of the tip revision with a previous revision

- 1 Select the item in the upper pane whose properties you want to compare.
You can select an item on any of the component tabs in the upper pane except the **Audit** tab.
- 2 Click the **History** tab in the lower pane and select any revision except the latest (tip) revision.
- 3 Right-click the selected revision and choose **Compare** ► **Properties**.

The **Compare Properties** window displays the properties of the tip revision and the selected historical revision in adjacent columns.

To compare the properties of two historical file revisions

- 1 Select the item in the upper pane whose historical properties you want to compare.
You can select an item on any of the component tabs in the upper pane except the **Audit** tab.
- 2 Click the **History** tab in the lower pane and select the two historical revisions whose properties you want to compare.
Use **CTRL+CLICK** to select the second revision.
- 3 Right-click the selection and choose **Compare** ► **Properties**.

The **Compare Properties** window displays the properties of the selected historical revision in adjacent columns.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Properties](#)
[Comparing Historical File Contents](#)
[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)

Configuring or Viewing Label Properties

View label properties include a name, description, frozen/unfrozen status, configuration, and build label status. Revision label properties include a name, description, and frozen/unfrozen status.

To display view or revision label properties for editing

- 1 Select the appropriate folder in the folder hierarchy tree.
- 2 Choose **View** ► **Labels**.
This displays the **Labels** dialog box.
- 3 Select a label on the **View** or **Revision** tabs and click **Properties**.
The **Edit Label** dialog box opens enabling you to modify the label name or description, and freeze or unfreeze the label.

To display folder label properties

- 1 Select the appropriate folder in the folder hierarchy tree.
- 2 Right-click the selected folder and choose **Labels**.
This displays the **Labels** dialog box showing all the labels attached to the folder.
- 3 Select a label from the list and click **Properties**.
A read-only **View Properties** dialog box opens enabling you to view the properties for the selected label.

To display label properties from the Label pane

- 1 Select an item in the upper pane.
- 2 Click the **Label** tab in the lower pane.
- 3 Right-click a label in the **Label** pane and choose **Properties**.
A read-only **View Properties** dialog box opens enabling you to view the properties for the selected label.

Related Concepts

[Labels](#)

[Folders](#)

[Overview of Views](#)

[Understanding Branching](#)

[Proper Use of Views](#)

Related Procedures

[Attaching Labels to Items](#)

[Creating View Labels](#)

[Creating Revision Labels](#)

[Reviewing and Moving Labels](#)

[Copying Revision Labels](#)

[Freezing or Unfreezing Labels](#)

[Promoting View Labels](#)

[Demoting View Labels](#)

[Configuring or Viewing Label Properties](#)

[Deleting Labels](#)

[Detaching Labels from Items](#)

Customizing Link Item Properties

You can view or modify folder and item properties directly from the **Link** pane.

To view or modify folder or item properties from a link

- 1 Select an item in the upper pane that links to a folder or another item.
- 2 Select the **Link** tab in the lower pane.
- 3 Right-click a link and choose **Item Properties** to display a **Properties** dialog box.
This dialog box displays information about the folder or item in the **Item** column in the **Link** pane.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Related Reference

[Project, Folder, and Item Properties](#)

Customizing Link Properties

You can view or modify link properties from the **Link** pane.

To view or modify link properties from a folder

- 1 Right-click the folder and choose [Properties](#).
- 2 Click the **Link** tab in the **Folder Properties** dialog box.
The **Link** pane displays all links for the selected folder.
- 3 Right-click a link and choose [Link Properties](#).
The **Link Properties** dialog box opens where you can view or modify certain properties or add a comment.

To view or modify link properties from an item

- 1 Select an item on one of the component tabs in the upper pane.
- 2 Select the **Link** tab in the lower pane.
The **Link** pane displays all links for the selected item.
- 3 Right-click a link in the **Link** pane and choose [Link Properties](#).
The **Link Properties** dialog box opens where you can view or modify certain properties or add a comment.

Related Concepts

[Links: Internal and External](#)
[Check-in and Check-out Operations](#)

Related Procedures

[Linking Items Internally or Externally](#)
[Linking Specific Revisions](#)
[Reviewing Linked Change Requests](#)
[Checking Linked Files In and Out](#)
[Selecting Linked Files](#)
[Deleting Links](#)
[Linking Files to Process Items](#)

Related Reference

[Project, Folder, and Item Properties](#)

Displaying Historical Properties

You can display the properties for a previous item revision using the **History** pane.

To compare the item properties of a historical revisions

- 1 Select the item in the upper pane whose historical properties you want to view.
You can select an item on any of the component tabs in the upper pane except the **Audit** tab.
- 2 Click the **History** tab in the lower pane.
- 3 Right-click the historical revision in the lower pane and choose **Properties**.

The item's **Properties** dialog box displays the properties of that revision.

Related Concepts

[Files](#)

[Folders](#)

Related Procedures

[Working with Properties](#)

[Comparing Historical File Contents](#)

[Working with Folders and Items](#)

[Managing Files](#)

[Managing Projects](#)

[Branching Operations](#)

[Linking and Unlinking Items](#)

[Filtering Data](#)

Emailing Item Properties

You can send a text representation of selected items (except files) as an email message, along with additional text. The information sent for each item includes the fields displayed in the upper pane. For items such as change requests, the item's properties, which are the same as its contents, are sent in the email. For files, only the properties can be sent. However, a shortcut to the item can be included.

Items are considered to have been sent by the application, not by you. Therefore, you may want to copy yourself on the email. Otherwise, you will not receive the message.

Note: If you set up a filter in the client and email an item, only the fields displayed by the filter are sent to the recipient.

To send item properties by email

- 1 Select a folder in the folder hierarchy, click a component tab, and select an item in the upper pane.
- 2 Do one of the following:
 - ◆ Click the **Send** button on the toolbar. (If you have selected a file component, this button does not appear.)
 - ◆ Right-click the selected item and choose **Send To**.

This opens the **Send To** dialog box.

- 3 Click **To** or **CC** to open a dialog box for selecting the primary or secondary email recipients.
Select the email recipients by moving the team member names from the **Available Users** to **Selected Users** list and click **OK**.
- 4 Type a **Subject**.
- 5 Optionally, check **Send A Copy To Myself** if you want to receive a copy of the email.
- 6 Optionally, check **Attach Item Shortcut** to include a shortcut to this specific item in the email.
- 7 Type any additional information in the **Add Text To The Mail Message** text box.
- 8 Click **Send Now** to send the message.

Unlike automatic email notification, this message will not display the word “notification” in the subject line.

Note: Do not confuse email messages sent by individuals with email notification messages automatically sent by the server. If your administrator has enabled email notification, you will automatically receive email messages notifying you about items for which you are responsible and topics for which you are listed as a recipient.

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Properties](#)
[Comparing Historical File Contents](#)
[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)


Using Embedded Property Editors

Embedded property editors allow you to:



- ◆ View, edit, and compare multiple StarTeam items at once.
- ◆ Copy values from one property editor to another.
- ◆ Continue to do other work while the embedded property editor is open, even if it has unsaved changes.

Note: StarTeam supports embedded editors for any combination of Alternative Property Editors (APEs), Layout Designer forms, and default property editors. However, embedded editing is not supported for APEs that do not implement the AbstractItemEditor interface; these APEs will be displayed in a property editor dialog.

To use embedded property editors


- 1 Click the  **Use Embedded Editor** icon on the toolbar.
The persistent default display mode for the selected item type will now be embedded editors instead of property editor dialogs.
- 2 To open an item, do one of the following:
 - ◆ Double-click the item.
 - ◆ Right-click the item and select **Properties** from the shortcut menu.


Each item opened in an embedded editor is displayed as a tab in the Embedded Editor pane below the Item pane. Each tab represents a unique item (and version) of the same or different Item type. You can view multiple versions of the same Item as well as Items of different types in the same Embedded Editor pane.


Once an embedded editor is open, use the  **Details Display** and  **Embedded Editor Display** icons to toggle back and forth between the Details pane and the Embedded Editor pane.


When an item is edited, asterisks will appear in the title bar and tab of the item's embedded property editor to indicate there are unsaved changes. On the right side of the title bar, all of the toolbar icons will also be enabled (the **Close** and **Close All** icons are always enabled). Each item's embedded property editor toolbar contains the following icons:


Embedded Property Editor Toolbar Icons


The  **Compare Properties** icon is enabled when there are unsaved changes to the item displayed in the embedded property editor. Click the icon to view the Compare Properties dialog and review changes made to the item during the current session. The original and new values for edited fields are displayed in bold type. Note that this icon is currently not available for an embedded APE editor.

The  **Save and Close** icon is enabled when there are unsaved changes to the item displayed in the embedded property editor. Click the icon to save the changes to the Item and close the item's embedded editor tab.

The  **Apply** icon is enabled when there are unsaved changes to the item displayed in the embedded property editor. Click the icon to save the changes to the Item and leave the item's embedded editor tab open.

The  **Reset** icon enabled when there are unsaved changes to the item displayed in the embedded property editor. Click the icon to discard changes made to the item during the current session and leave the item's embedded editor tab open.

The  **Close** icon is always enabled. Click this icon to close the item displayed in the embedded property editor. You will be prompted to save any changes to the item. You can also click the **X** on the tab to close the item.

The  **Close All** icon is always enabled. Click this icon to close all open embedded editors. You will be prompted to individually save each item that has unsaved changes.

To drag and drop items into the Details/Embedded Editor pane to view an embedded editor

- 1 Drag the first item(s) from the Items pane into the Details/Embedded Editor pane.
- 2 Drag any successive item(s) to the title bar of the active item's embedded editor tab.

Note: An item cannot be dropped onto an enabled toolbar on the right side of the title bar.

You can drag and drop an item into the Details/Embedded Editor pane whether or not embedded editors is the current display mode for the selected item type.

To toggle between full screen and standard view of the embedded editor

- 1 Double-click the title bar of the embedded property editor to view it in full screen mode.
- 2 Double-click the title bar of the property editor again to return to standard view.

Note: If you resize the Details/Embedded Editor pane, the Client will automatically display the split pane bars in the same position the next time you open the view window. This position can vary depending on whether you are viewing Details or Embedded Editors.

Viewing or Modifying Item Properties

This section explains how to use the standard properties dialog to edit item properties. Depending on how your team has set up the application, you may see a totally different dialog box called an alternate property editor (APE).

Every time the properties of an item are modified, a new revision of that item is created. If you modify a property, you should also create a revision comment explaining the modification using the **Revision Comment** tab.

To view or modify item properties

- 1 Click a folder in the folder hierarchy tree, or click one of the component tabs in the upper pane and select an item.
- 2 Do one of the following:
 - ◆ Right-click the folder or item and choose **Properties**.
 - ◆ Right-click an item on the **Reference** tab in the lower pane and choose **Item Properties**.
 - ◆ Click the **Properties** toolbar button.
- 3 Modify any of the property fields in the corresponding **Properties** dialog box that opens, then click **OK**.

Tip: You can force Java to not cache Windows file properties by placing the following option in the CPC stjava file:

```
-Dsun.io.useCanonCaches=false.
```

Related Concepts

[Files](#)
[Folders](#)

Related Procedures

[Working with Folders and Items](#)
[Managing Files](#)
[Managing Projects](#)
[Branching Operations](#)
[Linking and Unlinking Items](#)
[Filtering Data](#)
[Viewing or Modifying Item Properties](#)

Related Reference

[Change Request Properties](#)
[File Properties](#)
[Folder Properties](#)
[Requirement Properties](#)
[Task Properties](#)
[Topic Properties](#)

Creating Reports and Exporting Data

StarTeam provides the ability to create and customize reports, and to export data to various types of files.

In This Section

[Configuring the Report Output Path](#)

Describes how to configure where reports are saved.

[Creating Reports](#)

Describes how to create reports.

[Customizing Change Request Reports](#)

Describes how to customize a change request report.

[Customizing Fields Used In Report Templates](#)

Describes how to customize the fields in a report template.

[Customizing Report Templates](#)

Describes how to customize your report templates.

[Exporting Data](#)

Describes how to export data for use in other applications, such as spreadsheets.

[Generating Reports from a File Compare/Merge Session](#)

Describes the procedures for generating and printing reports from a File Compare/Merge.

[Printing Reports](#)

Describes how to print reports.

Configuring the Report Output Path

You can customize the location in which you store the reports you create. If desired, when you create a report, you can select an alternative location.

To configure the report output path

- 1 Select **Tools** ► **Personal Options** from the menu bar. The **Personal Options** dialog opens.
- 2 Select the **Workspace** tab.
- 3 Type or browse for a path for the **Report output path** text box. This path becomes the default location for all reports that you create using the application.

Note: The path you specify for **Output filename** in the **Reports** dialog overrides the default report output path specified in **Personal Options**.

- 4 Click **OK**.

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports](#)

Creating Reports

All reports that you create in the application show all or some portion of the data displayed in the upper pane. All reports are generated in .html format.

Note: You can also create reports using StarTeam Datamart. See the “StarTeam Datamart User Guide.pdf” for more information.

To create reports

- 1 Select a folder from the folder hierarchy.
- 2 Click a component tab.
- 3 Choose **Reports** from the component or context menu. The **Reports** dialog displays the **Available Reports** list box.
- 4 From the **Available Reports** list box, choose the type of the report you want to generate.
- 5 Do one of the following:
 - ◆ To include only the items selected on the upper pane, select the **Current Selection** option button.
 - ◆ To include all items displayed in the upper pane, select the **Select All** option button.
- 6 Type or browse to the path and report filename in the **Output file name** text box.
Be sure to use `.htm` or `.html` as the file extension.

Note: By default, the default report filename uses the convention
`<STReport><date><alphanumeric code>.html` (e.g.
`STReport2006-07-24T22-03-59Z.html`).
- 7 Type a name for your report in the **Report title** text box.
- 8 Click **Generate** to view the report on screen. Your Web browser opens and displays your report. In addition, the report is saved in the location specified in step 6 above.

Related Concepts

[Reports](#)

[Data Export with Datamart](#)

Related Procedures

[Customizing Report Templates](#)

[Configuring the Report Output Path](#)

[Printing Reports](#)

Customizing Change Request Reports

You can create a number of reports using the change request report features. This topic demonstrates an example of creating a report of change requests fixed during a certain time period.

To create a report of change requests fixed during a certain time period

- 1 Select the root folder in the **Folder Tree**.
- 2 Click the **Change Request** tab.
- 3 Choose **Change Request** ▶ **All Descendants**.
- 4 On the **Change Request** tab, display the **CR Number**, **Status**, **Modified Time**, and **Modified By** fields.
[Showing Fields in a Change Request](#)
- 5 Define the query that includes these fields and specifies a date range.
Specify a beginning **Modified Time**, and if the end date is not the current date, use an AND operator and specify an ending **Modified Time**.
[Selecting Change Requests Using a Query](#)
- 6 Sort and group the change requests, selecting **Status** from the **First By** drop-down list box and checking **Group By**.
[Sorting and Grouping Change Requests](#)
- 7 Select the **Status:Fixed** group.
- 8 Choose **Change Request** ▶ **Reports** to create a report showing the fixed change requests.

Related Concepts

[Change Requests](#)

Related Procedures

[Working with Change Requests](#)
[Customizing Change Request Filters](#)
[Filtering Data](#)

Related Reference

[Fields](#)
[Change Requests](#)

Showing Fields in a Change Request

You can select which fields are displayed for a change request.

To show specific fields in a change request

- 1 Right-click the column headers and select **Show Fields** from the context menu.
This opens the **Show Fields** dialog box.
- 2 Make sure the **CR Number**, **Entered By**, and any other appropriate fields are displayed in the **Show these fields in this order** list.
- 3 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Selecting Change Requests Using a Query

You can use a simple or complex query to limit the change requests displayed to those that fit specific criteria.

To select change requests using a query

- 1 Click **Change Request** tab in the upper pane.
- 2 Right-click a column header and choose **Queries** from the context menu.
- 3 Choose an existing query in the **Queries** dialog.
- 4 Click **Select** to apply it to the list of change requests.
- 5 Follow the steps in “Creating Queries” to define a new query.

Related Concepts

[Change Requests](#)
[Queries](#)

Related Procedures

[Customizing Change Request Reports](#)
[Creating Queries](#)
[Working with Change Requests](#)

Sorting and Grouping Change Requests

You can sort change requests by the data in a particular column or group the change requests.

To sort or group change requests

- 1 Click on a column header on the Change Requests tab to sort the change requests by the data in that column.

Note: If you want to sort or group the change requests first by the data in one column and then by the data in another column, see the following steps. You can sort or group the change requests in up to four levels of groupings.

- 2 To sort or group the change requests in multiple levels, right-click the column headers and select **Sort and Group** from the context menu. The **Sort and Group** dialog box opens.
- 3 From the **First By** drop-down list box, select a column title.
- 4 Optionally, group the change requests by the data in this column, select the **Group By** check box.

If you select the **Group By** check box, the change requests are grouped together in nested lists and you must drill down to view the change requests in each group. If you do not select the **Group By** check box, the change requests are all displayed on the **Change Requests** tab, sorted by your choices in **Sort and Group** dialog box.

Note: By default, the column data is sorted or grouped based on the internal key or order. You can use the **Sort Options** button and choose to sort or group the data based on the text and optionally, case sensitivity.

- 5 Repeat steps three and four to define up to four levels of sort orders or groupings.
- 6 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Customizing Fields Used In Report Templates

You can change what fields appear in a report by changing the fields specified in the report's template. Fields must be specified in the templates using their internal identifier. This is usually different from their display name (which is used as the column header, the name displayed in a pane, and the name in the report).

Field names in the report template files are delimited by the characters “~~”. For example, `~~ReportTitle~~` appears in all of the `x.Title` report templates. In the Cross-Platform Client, whatever you type as the name of the report in the **Reports** dialog becomes the title of the report.

Tip: To see all field display names, right-click the column header in the upper pane and choose **Show Fields** from the context menu. To find the internal field name for a field, go to the “Fields” reference topic listed below.

To customize the fields in a report template

- 1 Open the report template file in your favorite text editor.
- 2 Edit the file following standard HTML file conventions.
- 3 Save the file.

Note: If you want to create a new report template that has this new field, save the file with a new filename. All related report template files should be created with the same root filename.

For example, the `TopicsSummary.Group1` template contains the following lines:

```
<TD WIDTH=450><FONT SIZE="2">~~Title~~</FONT></TD><TD WIDTH=0></TD>
<TD WIDTH=100><FONT SIZE="2">~~CreateUserID~~</FONT></TD><TD WIDTH=0></TD>
<TD WIDTH=120><FONT SIZE="2">~~CreatedTime~~</FONT></TD><TD WIDTH=0></TD>
```

If you wanted to add the `Read Status` field to this report, you would need to know that its internal identifier is `ReadStatus`. Then you might add the following line to the template:

```
<TD WIDTH=120><FONT SIZE="2">~~ReadStatus~~</FONT></TD><TD WIDTH=0></TD>
```

Remember that, even if you specify a field in a template, that field is skipped if it does not also appear as a column in the project view window's upper pane. You might want to create filters for specific reports prior to creating the report.

Related Concepts

[Reports](#)

Related Procedures

[Customizing Report Templates](#)

[Creating Filters](#)

Related Reference

[Fields](#)

Customizing Report Templates

The application allows you to customize report templates. The templates are located in the folder you designated during installation. If you used the default path for a Windows installation, the Reports folder is `c:\Program Files\Borland\StarTeam Cross-Platform Client 2009\samples\details-templates\`. Be aware that different clients and client installations on different operating systems will probably have different installation folders. You may need to put templates in more than one location. All the templates are in HTML format.

While creating custom templates, keep these things in mind:

- ◆ A report typically consists of multiple template files, each using the naming convention `<ComponentReportname.Purpose>`. For example, the Default report provided for the Change Request component has five template files: `ChangeDefault.Name`, `ChangeDefault.Title`, `ChangeDefault.GrpInfo`, `ChangeDefault.Group1`, and `ChangeDefault.EndReport`.
- ◆ A double tilde (~~) precedes and follows field names in reports. For example, in the `ChangeDefault.Title` and other “Title” template files, the field name `~~ReportTitle~~` is used.

To customize a report template

- 1 From the Reports folder, copy all report template files with the same root name to the Reports folder using a different root name.

For example, if the Default report for the file component is the most similar to the new report you want to create, copy all `FileDefault.x` template files using a name like `FileMYREPORT.x`. In this case, you would create new files named `FileMYREPORT.Name`, `FileMYREPORT.Title`, and so on. In the Cross-Platform Client, the new report would appear in the **Report** dialog box with the name `MYREPORT`, along with the original `Default` report.

- 2 Open and edit the new report template files in any text editor or HTML editor.

Tip: A simple and easy method of creating and editing templates is to use Microsoft Word 97 or later, which includes automated HTML file generation.

- 3 Save the report templates you edited.

Related Concepts

[Reports](#)

Related Procedures

[Customizing Fields Used In Report Templates](#)

Related Reference

[Report Templates](#)

Exporting Data

You are limited to exporting 60 columns of data that display in the upper pane. Databases have limitations as well. Verify that the application in which you intend to use the exported data can accept all 60 columns.

Exported data can be imported into Microsoft Access, Microsoft Excel, or other applications.

To export data that can be imported into other applications

- 1 Select a folder in the **Folder Tree** and a component tab in the upper pane.

If you are exporting specific items in the upper pane, also select the specific items.

- 2 Right-click in the upper pane and choose **Advanced** ► **Export** from the context menu.

Note: For the Audit component only, select **Export** from the context menu.

This opens the **Export** dialog box.

- 3 Move the fields you want to export from the **Available fields** list to the **Show these fields in this order** list. Move them in the order you want them presented in the target application.

- 4 Specify which items to export in the **Export for** group box.

You can export selected items or all items in the upper pane.

- 5 Select the desired export options in the **Options** group box.

- 6 Type a path and filename in the **Output file name** text box, or browse to an existing file to overwrite it.

Note: Use `.txt` as the filename extension, or another extension which the target application can import. By default, the file is assigned a `.txt` extension.

- 7 Click **OK** to generate the export file.

- 8 Open the target application and import the exported file.

Print a report if desired.

Related Concepts

[Data Export with Datamart](#)

Related Reference

[Export Dialog Box Options](#)

Generating Reports from a File Compare/Merge Session

This procedure describes how to generate a difference report from File Compare/Merge session. It assumes that you are in a compare/merge session and want to generate a report of your changes before you commit them and close the session.

To generate a report from File Compare/Merge

- 1 Right-click in a File Compare/Merge pane and choose **Generate Report** to open the **Report** dialog box.

Note: If you are in the main or standalone File Compare/Merge window, you can use the **Generate Report** toolbar button.

- 2 Choose a report generator (transformer) from the drop-down list in the **Report** dialog box.
- 3 Optionally, when available, modify the report parameters to customize the report appearance.
- 4 Click **Next** to preview the report.
- 5 Click **Save** and browse to the target location for the report.
Choose a file extension that is valid for the output type (format) of the transformer you selected, and click **Save**.
- 6 Click **Close** in the **Report** dialog box.

Note: To print your report, open it in an appropriate application such as a text editor or a browser, and print.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)
[Setting File Compare/Merge Options](#)
[Comparing a Local File with a Repository File](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Generating Reports from a File Compare/Merge Session](#)
[Merging a Local File with the Tip Revision](#)
[Merging Folders](#)

Printing Reports

To print a report

- 1 Create the report and display it in the browser.
- 2 Right-click in the browser and choose **Print** from the context menu.
- 3 Set the print options in the **Print** dialog and then click the **Print** button.

Related Concepts

[Reports](#)

[Creating Reports](#)

Customizing Reports

You can customize change request and compare/merge reports, and modify the report templates and the fields in them.

In This Section

[Customizing Change Request Reports](#)

Describes how to customize a change request report.

[Customizing Compare and Merge Reports](#)

Describes how to use customized report transformers to generate reports from a compare/merge session.

[Customizing Fields Used In Report Templates](#)

Describes how to customize the fields in a report template.

[Customizing Report Templates](#)

Describes how to customize your report templates.

Customizing Change Request Reports

You can create a number of reports using the change request report features. This topic demonstrates an example of creating a report of change requests fixed during a certain time period.

To create a report of change requests fixed during a certain time period

- 1 Select the root folder in the **Folder Tree**.
- 2 Click the **Change Request** tab.
- 3 Choose **Change Request** ▶ **All Descendants**.
- 4 On the **Change Request** tab, display the **CR Number**, **Status**, **Modified Time**, and **Modified By** fields.
[Showing Fields in a Change Request](#)
- 5 Define the query that includes these fields and specifies a date range.
Specify a beginning **Modified Time**, and if the end date is not the current date, use an AND operator and specify an ending **Modified Time**.
[Selecting Change Requests Using a Query](#)
- 6 Sort and group the change requests, selecting **Status** from the **First By** drop-down list box and checking **Group By**.
[Sorting and Grouping Change Requests](#)
- 7 Select the **Status:Fixed** group.
- 8 Choose **Change Request** ▶ **Reports** to create a report showing the fixed change requests.

Related Concepts

[Change Requests](#)

Related Procedures

[Working with Change Requests](#)
[Customizing Change Request Filters](#)
[Filtering Data](#)

Related Reference

[Fields](#)
[Change Requests](#)

Showing Fields in a Change Request

You can select which fields are displayed for a change request.

To show specific fields in a change request

- 1 Right-click the column headers and select **Show Fields** from the context menu.
This opens the **Show Fields** dialog box.
- 2 Make sure the **CR Number**, **Entered By**, and any other appropriate fields are displayed in the **Show these fields in this order** list.
- 3 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Selecting Change Requests Using a Query

You can use a simple or complex query to limit the change requests displayed to those that fit specific criteria.

To select change requests using a query

- 1 Click **Change Request** tab in the upper pane.
- 2 Right-click a column header and choose **Queries** from the context menu.
- 3 Choose an existing query in the **Queries** dialog.
- 4 Click **Select** to apply it to the list of change requests.
- 5 Follow the steps in “Creating Queries” to define a new query.

Related Concepts

[Change Requests](#)
[Queries](#)

Related Procedures

[Customizing Change Request Reports](#)
[Creating Queries](#)
[Working with Change Requests](#)

Sorting and Grouping Change Requests

You can sort change requests by the data in a particular column or group the change requests.

To sort or group change requests

- 1 Click on a column header on the Change Requests tab to sort the change requests by the data in that column.

Note: If you want to sort or group the change requests first by the data in one column and then by the data in another column, see the following steps. You can sort or group the change requests in up to four levels of groupings.

- 2 To sort or group the change requests in multiple levels, right-click the column headers and select **Sort and Group** from the context menu. The **Sort and Group** dialog box opens.
- 3 From the **First By** drop-down list box, select a column title.
- 4 Optionally, group the change requests by the data in this column, select the **Group By** check box.

If you select the **Group By** check box, the change requests are grouped together in nested lists and you must drill down to view the change requests in each group. If you do not select the **Group By** check box, the change requests are all displayed on the **Change Requests** tab, sorted by your choices in **Sort and Group** dialog box.

Note: By default, the column data is sorted or grouped based on the internal key or order. You can use the **Sort Options** button and choose to sort or group the data based on the text and optionally, case sensitivity.

- 5 Repeat steps three and four to define up to four levels of sort orders or groupings.
- 6 Click **OK**.

Related Concepts

[Change Requests](#)

Related Procedures

[Customizing Change Request Reports](#)

[Creating Queries](#)

[Working with Change Requests](#)

Customizing Compare and Merge Reports

This procedure describes how to use customized report transformer. (.xsl stylesheets). You need to understand how to create an .xsl stylesheet to use as a template for creating a report transformer, or you need access to one created by someone else.

To use a custom .xsl transform stylesheet

- 1 Choose **Tools** ► **Options** in the File Compare/Merge window and expand the **File Comparison** or **Folder Comparison** node.
Select the **Report Transformers** page.
- 2 Click **Add** to open the **Report Transformer Configuration** dialog box.
- 3 Type a name for the report transformer, and choose the type of report you want to generate from the **Output Type** drop-down list.
- 4 Click **Select XSL Template** and browse to your custom .xsl stylesheet
You can modify the stylesheet in the edit box of the **Report Transformer Configuration** dialog box. If you do modify the file, click **Save XSL Template** to save it with an .xsl extension.
- 5 Click **OK** to close the **Report Transformer Configuration** and return to the File Compare/Merge Options dialog box.
Your new report transformer is added to the list of available report transformers in the Report dialog box that opens when you generate a report in a View Compare/Merge session.

Related Procedures

[Comparing and Merging Views](#)

[Comparing Selected Items in the Source and Target View](#)

[Generating Reports from a File Compare/Merge Session](#)

Customizing Fields Used In Report Templates

You can change what fields appear in a report by changing the fields specified in the report's template. Fields must be specified in the templates using their internal identifier. This is usually different from their display name (which is used as the column header, the name displayed in a pane, and the name in the report).

Field names in the report template files are delimited by the characters “~~”. For example, `~~ReportTitle~~` appears in all of the `x.Title` report templates. In the Cross-Platform Client, whatever you type as the name of the report in the **Reports** dialog becomes the title of the report.

Tip: To see all field display names, right-click the column header in the upper pane and choose **Show Fields** from the context menu. To find the internal field name for a field, go to the “Fields” reference topic listed below.

To customize the fields in a report template

- 1 Open the report template file in your favorite text editor.
- 2 Edit the file following standard HTML file conventions.
- 3 Save the file.

Note: If you want to create a new report template that has this new field, save the file with a new filename. All related report template files should be created with the same root filename.

For example, the `TopicsSummary.Group1` template contains the following lines:

```
<TD WIDTH=450><FONT SIZE="2">~~Title~~</FONT></TD><TD WIDTH=0></TD>
<TD WIDTH=100><FONT SIZE="2">~~CreateUserID~~</FONT></TD><TD WIDTH=0></TD>
<TD WIDTH=120><FONT SIZE="2">~~CreatedTime~~</FONT></TD><TD WIDTH=0></TD>
```

If you wanted to add the `Read Status` field to this report, you would need to know that its internal identifier is `ReadStatus`. Then you might add the following line to the template:

```
<TD WIDTH=120><FONT SIZE="2">~~ReadStatus~~</FONT></TD><TD WIDTH=0></TD>
```

Remember that, even if you specify a field in a template, that field is skipped if it does not also appear as a column in the project view window's upper pane. You might want to create filters for specific reports prior to creating the report.

Related Concepts

[Reports](#)

Related Procedures

[Customizing Report Templates](#)

[Creating Filters](#)

Related Reference

[Fields](#)

Customizing Report Templates

The application allows you to customize report templates. The templates are located in the folder you designated during installation. If you used the default path for a Windows installation, the Reports folder is `c:\Program Files\Borland\StarTeam Cross-Platform Client 2009\samples\details-templates\`. Be aware that different clients and client installations on different operating systems will probably have different installation folders. You may need to put templates in more than one location. All the templates are in HTML format.

While creating custom templates, keep these things in mind:

- ◆ A report typically consists of multiple template files, each using the naming convention `<ComponentReportname.Purpose>`. For example, the Default report provided for the Change Request component has five template files: `ChangeDefault.Name`, `ChangeDefault.Title`, `ChangeDefault.GrpInfo`, `ChangeDefault.Group1`, and `ChangeDefault.EndReport`.
- ◆ A double tilde (~~) precedes and follows field names in reports. For example, in the `ChangeDefault.Title` and other "Title" template files, the field name `~~ReportTitle~~` is used.

To customize a report template

- 1 From the Reports folder, copy all report template files with the same root name to the Reports folder using a different root name.

For example, if the Default report for the file component is the most similar to the new report you want to create, copy all `FileDefault.x` template files using a name like `FileMYREPORT.x`. In this case, you would create new files named `FileMYREPORT.Name`, `FileMYREPORT.Title`, and so on. In the Cross-Platform Client, the new report would appear in the **Report** dialog box with the name `MYREPORT`, along with the original `Default` report.

- 2 Open and edit the new report template files in any text editor or HTML editor.

Tip: A simple and easy method of creating and editing templates is to use Microsoft Word 97 or later, which includes automated HTML file generation.

- 3 Save the report templates you edited.

Related Concepts

[Reports](#)

Related Procedures

[Customizing Fields Used In Report Templates](#)

Related Reference

[Report Templates](#)

Creating Charts

This section contains tasks related to creating charts.

In This Section

[Choosing the Chart Type](#)

Describes how to choose a chart type in the **Chart** window.

[Configuring Chart Colors](#)

Describes how to change the color of series data in a chart.

[Customizing Chart Titles](#)

Describes how to change the titles on charts.

[Exporting a Chart as an Image](#)

Describes how to export a chart as an image.

[Generating Correlation Charts](#)

Describes how to create a Correlation chart from data in the upper pane.

[Generating Distribution Charts](#)

Describes how to create a Distribution chart from data in the upper pane.

[Generating Simple Charts](#)

Describes how to generate a Simple chart from data in the upper pane.

[Generating Time-series Charts](#)

Describes how to create a Time-series chart from data in the upper pane.

[Viewing Charts](#)

Describes various ways to view data in a chart.

Choosing the Chart Type

Once you have generated a chart, the **Chart** window opens displaying a default chart type. You can choose a different chart type.

To change the chart type

- 1 Generate your chart.
- 2 Choose a different chart type from the drop-down list on the toolbar in the **Chart** window.

Related Procedures

- [Generating Simple Charts](#)
- [Generating Distribution Charts](#)
- [Generating Correlation Charts](#)
- [Generating Time-series Charts](#)
- [Viewing Charts](#)
- [Exporting a Chart as an Image](#)
- [Configuring Chart Colors](#)
- [Customizing Chart Titles](#)

Related Reference

- [Charts](#)

Configuring Chart Colors

You can control the color of series data in the StarTeam **Chart** window.

To change the colors of series data

- 1 Click **Edit Colors** on the **Chart** window toolbar.

The **Edit Colors** dialog box opens.

- 2 Check **Use Custom Colors**.
- 3 Select a series from the list and click **Edit**,

Note: You can edit only one series at a time.

- 4 Choose the color for the series in the **Select Color** dialog box using the swatches of color or the HSB or RGB values.
- 5 Click **OK** to return to the **Edit Color** dialog box.
- 6 Select another series to edit its color.
- 7 Click **OK** to apply your changes and exit the **Edit Colors** dialog box.

Related Procedures

[Generating Simple Charts](#)
[Generating Distribution Charts](#)
[Generating Correlation Charts](#)
[Generating Time-series Charts](#)
[Viewing Charts](#)
[Sorting and Grouping Data](#)
[Choosing the Chart Type](#)
[Exporting a Chart as an Image](#)
[Customizing Chart Titles](#)

Related Reference

[Charts](#)

Customizing Chart Titles

When you create a chart, you can insert titles for the top, left, right, or bottom of the chart. These titles can serve as overall titles, or names for the axes.

To create or edit titles for your chart

- 1 Click **Edit Titles** on the **Chart** window toolbar.
- 2 Type or edit the contents of the **Top**, **Left**, **Right**, or **Bottom** text boxes.
- 3 Click **Apply** to make the changes and keep the dialog box open, or click **OK** to close the dialog box.

Related Procedures

- [Generating Simple Charts](#)
- [Generating Distribution Charts](#)
- [Generating Correlation Charts](#)
- [Generating Time-series Charts](#)
- [Viewing Charts](#)
- [Sorting and Grouping Data](#)
- [Choosing the Chart Type](#)
- [Exporting a Chart as an Image](#)
- [Configuring Chart Colors](#)

Related Reference

- [Charts](#)

Exporting a Chart as an Image

You can export a chart as a [.jpg](#) image from the **Chart** window

To export/save a chart as an image

- 1 Click the **Save Chart As** button on the **Chart** window toolbar.
- 2 Type a name for the file in the **Save As** dialog box.
- 3 Browse to or type a path to the target location for the [.jpg](#) file.
- 4 Click **Save**.

Related Procedures

[Viewing Charts](#)

[Generating Simple Charts](#)

[Generating Distribution Charts](#)

[Generating Correlation Charts](#)

[Generating Time-series Charts](#)

[Choosing the Chart Type](#)

[Configuring Chart Colors](#)

[Customizing Chart Titles](#)

Related Reference

[Charts](#)

Generating Correlation Charts

This procedure explains how to generate a *Correlation* chart from the item data in the upper pane. A Correlation chart displays as a scatter chart, based on the fields you specify for the x-axis and y-axis.

You can filter out data in the upper pane to display only the data that you want to include in your chart. In addition, you can select specific items from the filtered data to include in your chart. A maximum of 60 fields can be displayed in the upper pane.

Note: You can only generate a Distribution or Time-series chart for audit entries

To generate a Correlation chart

- 1 Select the tab in the upper pane containing the items you want to chart: **File**, **Change Request**, **Requirement**, **Task**, **Topic**, **Audit**, or **Child Folder**.
- 2 Optionally, select the specific items you want to chart.
- 3 Right-click in the upper pane and choose **Charts** ► **Correlation** to open the **Correlation Chart** dialog box.
- 4 Type a name for the chart in the **Chart Name** text box.
- 5 Select one axis label from the **x-coordinates** list box, and one from the **y-coordinates** list box.

Note: These coordinate list boxes display the names of the fields that are displayed in the upper pane that can be used as axes. A maximum of sixty fields can be displayed in the upper pane.

- 6 Select a printer page orientation: **Portrait** or **Landscape**.
- 7 Click **OK** to display the chart.

Note: If the chart has too much data on it to be readable, increase the size of the chart window, or decrease the number of items in the chart.

Related Procedures

[Viewing Charts](#)
[Sorting and Grouping Data](#)
[Choosing the Chart Type](#)
[Exporting a Chart as an Image](#)
[Configuring Chart Colors](#)
[Customizing Chart Titles](#)

Related Reference

[Charts](#)

Generating Distribution Charts

You can filter out data in the upper pane to display only the data that you want to include in your chart. In addition, you can select specific items from the filtered data to include in your chart. A maximum of 60 fields can be displayed in the upper pane.

This procedure explains how to generate a *Distribution* chart of items grouped in the upper pane. A Distribution chart displays in the form of a pie chart. Each wedge indicates what fraction of the whole a group represents.

Note: You can only generate a Distribution or Time-series chart for audit entries

To generate a Distribution chart

- 1 Select the tab in the upper pane containing the items you want to chart: **File**, **Change Request**, **Requirement**, **Task**, **Topic**, **Audit**, or **Child Folder**.
- 2 Sort and/or Group the data on the selected tab in the upper pane.
- 3 Right-click in the upper pane and choose **Charts** ► **Distribution**.
This opens the **Distribution Chart** dialog box.
- 4 Type a name for the chart in the **Chart Name** text box.
- 5 Select which data to use for generating the chart.
If you have selected specific items in the upper pane to chart, select **Current Selection**. Otherwise, select **Select All**.
- 6 Select a printer page orientation: **Portrait** or **Landscape**.
- 7 Click **OK** to display the chart.

Note: If the chart has too much data on it to be readable, increase the size of the chart window, or decrease the number of items in the chart.

Related Procedures

[Viewing Charts](#)
[Sorting and Grouping Data](#)
[Choosing the Chart Type](#)
[Exporting a Chart as an Image](#)
[Configuring Chart Colors](#)
[Customizing Chart Titles](#)

Related Reference

[Charts](#)

Generating Simple Charts

This procedure explains how to generate a *Simple* chart from the item data displayed in the upper pane. A Simple chart displays in a column format by default. However, the chart utility allows you to display simple charts in a variety of formats.

You can filter out data in the upper pane to display only the data that you want to include in your chart. In addition, you can select specific items from the filtered data to include in your chart. A maximum of 60 fields can be displayed in the upper pane.

Note: You can only generate a Distribution or Time-series chart for audit entries

To generate a Simple chart

- 1 Select the tab in the upper pane containing the items you want to chart: **File**, **Change Request**, **Requirement**, **Task**, **Topic**, **Audit**, or **Child Folder**.
- 2 Optionally, select the specific items you want to chart.
- 3 Right-click in the upper pane and choose **Charts** ► **Simple**.
This opens the **Simple Chart** dialog box.
- 4 Type a name for the chart in the **Chart Name** text box.
- 5 Select one or more fields from the **Series** list box.
Use **CTRL+CLICK** or **SHIFT+ARROW** to multiple select the fields. The dialog box lists the fields in the upper pane that can be charted. A maximum of sixty fields can be displayed in the upper pane.

Note: The Series list box can only contain fields that are number based. To add fields to the Series list box, you must include a number-based field in the view/tab you want to chart.
- 6 Select which data to use for generating the chart.
If you have selected specific items in the upper pane to chart, select **Current Selection**. Otherwise, select **Select All**.
- 7 Select a printer page orientation: **Portrait** or **Landscape**.
- 8 Click **OK** to display the chart.

Note: If the chart has too much data on it to be readable, increase the size of the chart window, or decrease the number of items in the chart.

Related Procedures

[Viewing Charts](#)
[Sorting and Grouping Data](#)
[Choosing the Chart Type](#)
[Exporting a Chart as an Image](#)
[Configuring Chart Colors](#)
[Customizing Chart Titles](#)

Related Reference

[Charts](#)

Generating Time-series Charts

This procedure explains how to generate a *Time-series* chart from the item data displayed in the upper pane. A Time-series chart displays as a line chart showing the number of items that have the same day, week, or month in the specified time/date field.

You can filter out data in the upper pane to display only the data that you want to include in your chart. In addition, you can select specific items from the filtered data to include in your chart. A maximum of 60 fields can be displayed in the upper pane.

To generate a Time-series chart

- 1 Select the component tab in the upper pane containing the items you want to chart.

You can generate time-series charts for all item types.

Note: A time-series chart must be based on a time/date field. If the upper pane displays no time/date fields, the application displays an error message. To avoid this error, add a time/date field to the columns in the upper pane.

- 2 Optionally, select the specific items you want to chart.
- 3 Right-click in the upper pane and choose **Charts** ► **Time-series** to open the **Time-series Chart** dialog box.
- 4 Type a name for the chart in the **Chart Name** text box.
- 5 Select a **Tracking Interval**: **Daily**, **Weekly**, or **Monthly**
- 6 Select one or more items from the **Time Series** list box.
- 7 Check any additional options you want.

You can specify a date range (**Limit To Period Of Time**), display accumulated items (**Cumulative Totals**), or include all date ranges in the chart (**Include Non-Represented Dates**).

- 8 Select which data to use for generating the chart.

If you have selected specific items in the upper pane to chart, select **Current Selection**. Otherwise, use **Select All**.

- 9 Select a printer page orientation: **Portrait** or **Landscape** then click **OK** to display the chart.

Note: If the chart has too much data on it to be readable, increase the size of the chart window, or decrease the number of items in the chart.

Related Procedures

[Viewing Charts](#)
[Sorting and Grouping Data](#)
[Choosing the Chart Type](#)
[Exporting a Chart as an Image](#)
[Configuring Chart Colors](#)
[Customizing Chart Titles](#)

Related Reference

[Charts](#)

Viewing Charts

The StarTeam **Chart** window displays the majority of charts by default in 3D, but you also have the option to toggle the 3D/2D view. You can zoom in and out, and rotate a chart on its x-y-z axes for all charts that you can view in 3D. The majority of charts shown in 2D do not allow for zoom or rotate operations.

Other options you have in viewing charts are to show a legend, and to display a horizontal or vertical grid.

The procedures below include the following:

- ◆ Viewing a Chart in 3D or 2D
- ◆ Zooming in and out on a Charts
- ◆ Rotating a Chart
- ◆ Displaying a Legend on a Chart
- ◆ Displaying a Horizontal of Vertical Grid on a Chart

To toggle between a 3D and a 2D chart

- 1 Generate a chart and display it in the **Chart** window.
- 2 Click the **Toggle 3D View** button on the **Chart** window toolbar.
- 3 To return to a 3D chart, click the **Toggle 3D View** button again.

To zoom in on a chart

- 1 Generate a chart and display it in the **Chart** window.
- 2 Be sure that the chart type selected is in a view that allows for zooming.

Note: To see which charts allow for zooming, refer to the reference topic "Default Chart Views and Zoom/Rotate Capabilities" in the links below.

- 3 While holding down the **CTRL** key, click and hold down the left mouse button and move your mouse from left to right to zoom in and out on the chart.

To rotate a chart

- 1 Generate a chart and display it in the **Chart** window.
- 2 Be sure that the chart type selected is in a view that allows for rotating.

Note: To see which charts allow for rotating, refer to the reference topic "Default Chart Views and Zoom/Rotate Capabilities" in the links below.

To display the chart data on which the table is based

- 1 Generate a chart and display it in the **Chart** window.
- 2 Click **Toggle Table** on the **Chart** window toolbar.

The chart data displays in a matrix that shows the legend entries vertically and the charted values for each of the file entries horizontally.

- 3 Click **Toggle Table** again, to return to the chart display.

To display a legend on a chart

- 1 Generate a chart and display it in the **Chart** window.
- 2 Click **Toggle Legend** on the **Chart** window toolbar.
Since the default is to display the legend, this hides the legend.
- 3 To display the legend, click **Toggle Legend** again.

To display a horizontal or vertical grid on a chart

- 1 Generate a chart and display it in the **Chart** window.
- 2 Click the **Horizontal Grid** button or the **Vertical Grid** button on the toolbar.
Since the horizontal and vertical grid lines are displayed by default, this action hides the grids.
- 3 Click **Horizontal Grid** or **Vertical Grid** again to add the grid lines back to the chart display.

Related Procedures

[Generating Simple Charts](#)
[Generating Distribution Charts](#)
[Generating Correlation Charts](#)
[Generating Time-series Charts](#)
[Sorting and Grouping Data](#)
[Choosing the Chart Type](#)
[Exporting a Chart as an Image](#)
[Configuring Chart Colors](#)
[Customizing Chart Titles](#)

Related Reference

[Charts](#)
[Default Chart Views and Zoom/Rotate Capabilities](#)

Accessing Projects and Items with Shortcuts and URLs

This section contains tasks related to using shortcuts and URLs to access projects and items.

In This Section

[Copying and Opening URLs](#)

Describes how to copy and open a URL to a StarTeam project, view, folder, or item.

[Creating Shortcuts](#)

Describes how to save a shortcut to an item.

Copying and Opening URLs

The StarTeam client can open URL shortcut links to projects, views, folders, Not-in-View folders, and items. This enables you to quickly access specific locations in a project. You can also copy a URL to the clipboard, which allows you to easily move its contents to an appropriate application, such as an email client or a document.

Like other URLs, StarTeam URLs include the name of the server for the connection. In some organizations, StarTeam servers may be reached from both the Internet and the corporate intranet. In such cases, a server may have two different IP addresses. If you configure the server list to reference a server by its IP address, rather than its DNS name, then any URLs generated by the client will work only from the network on which that IP address exists.

URL displays can be changed in your **Personal Options** settings.

Note: StarTeam shortcuts use the file extension `.stx`. For shortcuts to saved View Compare/Merge sessions, it uses `.vcmx` and `.vcms`.

To copy a URL to the clipboard

- 1 Select one or more items in the upper pane.
- 2 Right-click the selected items and choose **Copy URL to clipboard**.

This action places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.

- 3 Paste the URL to the application of choice.

Note: Copying a URL to the clipboard is equivalent to dragging an item or items from the list pane or folder tree onto an application. Not all applications support pasting the HTML representation, although Word, Excel, and Outlook do support HTML data.

To open a URL

- 1 Choose **Project** ► **Open StarTeam URL**.
- 2 Type a valid URL to a project, view, folder, or item in the **Open StarTeam URL** dialog box.

For example: `starteam:/hostname:49201/myproject`.

- ◆ If the URL is a reference to a project, the default view of the project opens.
- ◆ If the URL is a reference to a view or folder, then the view or folder opens.
- ◆ If the URL is a reference to an item, the item's view opens, the item's parent folder is selected in the folder tree, the item type is selected, and the item itself is selected in the item list or tree on the upper pane.

Tip: Double-click a StarTeam URL link in an email or other application to open StarTeam to the specified project, view, or item. In Windows Explorer, StarTeam links display a small StarTeam icon which you can double-click to open.

Related Procedures

[Opening a Saved or Exported View Compare/Merge Session](#)

Creating Shortcuts

For easy access to items that you are tracking, you can save shortcuts to them on your desktop. Opening a shortcut starts the application, opens the project view in the configuration it had when the shortcut was created, and displays the **Properties** dialog box for the item.

To save a shortcut to an item

- 1 Select a folder from the **Folder Tree**.
- 2 Click the appropriate component tab.
- 3 Right-click an item in the upper pane and choose **Save Shortcut**.
This opens the **Save As** dialog box.
- 4 Type a name for the shortcut in the **File Name** text box if you do not want to use the default name.
Be sure to keep the `.stx` extension.
- 5 Select a location, usually your desktop, for storing the shortcut and click Save.

Note: To open the item from the shortcut you saved, double-click the shortcut file (`.stx`).

Related Concepts

[Overview of Projects](#)

Related Procedures

[Adding Files to Projects](#)

[Saving Projects as Shortcuts](#)

[Opening Projects with Shortcuts](#)

Reference

This section contains reference information.

In This Section

[Access Rights and Privileges](#)

This section contains reference topics related to access rights and privileges.

[File Compare/Merge Reference](#)

Reference topics for File Compare/Merge.

[View Compare/Merge](#)

Contains reference topics related to View Compare/Merge in StarTeam.

[Cross-Platform Client Menus](#)

This section contains reference topics related to the menu commands found in the StarTeam Cross-Platform Client.

[Customization](#)

This section contains reference topics related to customization.

[Keyboard Shortcuts](#)

This section contains reference topics with tables of keyboard shortcuts.

[Project, Folder, and Item Properties](#)

This section contains detailed information about the properties for all types of items.

[Change Requests](#)

This section contains reference topics related to change requests.

[Charts](#)

This section contains reference topics related to charts.

[Exporting](#)

Describes the options for exporting items.

[Fields](#)

Detailed lists of the fields found in the upper pane on each of the component tabs, and relational operators used in queries

[File Status](#)

Describes the status of files and the effect of file status on check-in and check-out operations.

[Filters Reference Topics](#)

This section contains reference topics related to filters.

[Links](#)

This section contains UI overview topics related to Links.

[Queries](#)

This section contains reference topics related to queries.

[Reports](#)

This section contains reference topics related to reports and report templates.

[Personal Options Settings](#)

This section contains the documentation that explains all the settings in the Personal Options dialog box.

[Views](#)

Provides reference information on creating and configuring views.

[Audit Log Events](#)

Lists events provided in the audit log.

[Table of Common Operations](#)

Summarizes the operations that can be performed by each StarTeam component.

[Table of StarTeam Keywords](#)

Table of keywords for use in keyword expansion.

Access Rights and Privileges

This section contains reference topics related to access rights and privileges.

In This Section

[Group Privileges](#)

Describes the privileges assigned to a group.

[Server Access Rights](#)

Describes server-level access rights.

[Project Access Rights](#)

Describes generic access rights for projects.

[View Access Rights](#)

Describes access rights for views.

[Folder Access Rights](#)

Describes access rights for folders.

[Child Folder Access Rights](#)

Describes access rights for child folders.

[File Access Rights](#)

Describes file access rights.

[Generic Item Access Rights](#)

Describes generic item access rights available from the File nodes in the Project, View, Folder, and File Access Rights dialog boxes.

[Promotion State Access Rights](#)

Describes access rights for promotion states.

[Component Access Rights](#)

Describes access rights for components.

[Component-level Filter Access Rights](#)

Describes access rights for component-level filters.

[Individual Filter Access Rights](#)

Describes access rights for individual filters.

[Component-level Query Access Rights](#)

Describes access rights for component-level queries.

[Individual Query Access Rights](#)

Describes access rights for individual queries.

[Change Package Access Rights](#)

Describes the Change Package item access rights available from the Project and View menu Access Rights menu item.

[External Link Access Rights](#)

Describes the External Link item access rights available from the Project and View menu Access Rights menu item.

Group Privileges

The privileges assigned to a group may allow members of that group to access objects and perform operations that they are otherwise not allowed to do. In other words, privileges override the access rights settings.

If you select User Manager from the Server Administration dialog, you will notice that the server configuration comes with some default groups: All Users, Administrators, System Managers, and Security Administrators. The default user named Administrator belongs to both the Administrators and the Security Administrators groups. By default, the Administrators group has all group privileges. Also by default, other groups have none of these privileges.

All members of a group have the same privileges on every project managed by this server configuration. The privileges apply to all levels equally: projects, views, folders, and items within folders. If users belong to more than one group, they have the maximum amount of privileges, regardless of which group provides them with those privileges.

This privilege...	Allows a group to...
See object and its properties	See all projects, views, folders, items, and their properties. This privilege overrides the similarly named access right found in the Generic Object Rights in the Access Rights dialogs.
Modify object properties	Modify the properties of any projects, views, folders, or items. This privilege overrides the similarly named access right found in the Generic Object Rights in the Access Rights dialogs.
Delete object	Delete any projects, views, folders, or items. This privilege overrides the similarly named access right found in the Generic Object Rights in the Access Rights dialogs.
Purge object (delete permanently) Change object access right	This privilege is not supported at this time. Change access rights for any projects, views, folders, or items. This privilege overrides the similarly named access right found in the Generic Object Rights in the Access Rights dialogs.
Create object and place it in a container	Create new objects and put them in containers. When this privilege is set, the group can add new views to a project, new folders to a view, and new folders and items to a folder. This privilege overrides the similarly named access right found in the Generic Object Rights in the Access Rights dialogs. It does not override the server-level access right that allows users to create projects.
Grant all specific class-level rights for all classes of objects	Perform any operation not covered by the preceding privileges. For example, this privilege allows group members to check out files, break locks, perform linking operations, and perform labeling operations. This privilege overrides some of the access rights found in the Generic Object Container Rights and all of the access rights in the <item>-specific Rights in the Access Rights dialog.

Related Reference

[Access Rights and Privileges](#)

Server Access Rights

The server-level rights you assign to users and groups authorize them to perform specific operations in a particular server configuration. One of the options determines who can and who cannot create projects when the server configuration is running. Server rights can be assigned only when a server is running.

By default, the Administrators group is assigned all project and Server rights. By default, the All Users group has the rights to create projects and review the server configuration and the server log. The Server access rights are briefly described in the following table.

This access right...	Allows a user or group to...
View server log	Review, but not change, server log information.
View statistics and licensing information	Review, but not change, statistics information (StarTeam Server 5.4 and earlier). Create license usage files.
View server configuration	Review, but not change, the server configuration options.
Modify server configuration	Change the server configuration options.
Remotely administer server	Lock/unlock the server, restart the server from the client, shut down the server from the client, access the Start/Stop Conversion and Hive Manager vault buttons.
Administer user accounts	Add groups and users.
View system policy	Review, but not change, the password and logon failure options for the server configuration.
Modify system policy	Change the password and logon failure options for the server configuration.
Change server security settings	Set Server access rights. If you change this setting, be sure that you remain one of the users who can change access rights.
View security log	Review, but not change, server log information
StarDisk Operations	
Create new users	Add new users to sample project.
Replication Support	
Change user/operation time	Manipulate creation times and user names when using special clients, such as Notification Agent.
Project Operations	
Create projects	Create projects when the Server is running the server configuration.
Customizations	
Add/modify database schema	Create customized fields as item properties, or modify a field for an item that can be modified.
Component operations	
Administer component-level access rights	Designate the users and groups who can create and apply filters and queries for a specific component in the server configuration.

Related Reference

[Access Rights and Privileges](#)

Project Access Rights

The following table describes the generic object rights for a project. To display the Project Access Rights dialog, select the **Project** ► **Access Rights** command. The right to create a project is set as a Server access right.

This access right	Allows a user or group to...
See object and its properties	See this project and view its properties by selecting Project > Properties.
Modify properties	Change the properties for this project. The project properties that can be modified are name, description, keyword expansions settings, alternate property editor (APE) settings, process rules settings, requiring unlocked files to be read-only, and several settings that affect users (for example, requiring revision comments to be entered when a file is checked in).
Delete object	Delete this project from its server configuration.
Change object access rights	Change the access rights for this project. If you change this setting, be sure that you remain one of the users who can change access rights.

Related Reference

[Access Rights and Privileges](#)

View Access Rights

When you select the **View** ► **Access Rights** command to open the **View Access Rights** dialog, the rights shown are for the current view. The rights available from the **View** node are also available from the **View** node in the **Project Access Rights** dialog. In the latter case, the rights cover all views in the project rather than an individual view. It also include a container-level right that allows users or groups to create views for the project. This right is not available on the **View** node of the **View Access Rights** dialog box.

The following table describes the access rights that are available from the **View** node in the **Project Access Rights** dialog box. Most of these access rights also appear on the **View** node of the **View Access Rights** dialog box, but apply only to the current view.

This access right...	Allows a user or group to...
Generic Object Rights	
See object and its properties	Change view properties. View properties that can be modified are the view's name, description, working folder (also the root folder's working folder), branch setting for shared items, and file status repository setting.
Modify properties	Modifies the view properties.
Delete object	Deletes the object from the view.
Change object access rights	Changes the access rights of the selected object in the view.
View-Specific Rights	
Create view labels	Creates view labels. These labels will be automatically attached to the folders and items in the view. Users with this right but not the right to attach labels can still create labels.
Modify view labels	Changes the properties of view labels. For example, this right allows a user to freeze labels so that they cannot be adjusted
Delete view labels	Deletes view labels. This action automatically detaches the view labels from the folders and items that had the labels. Users with this right but not the right to detach view labels can still delete view labels.
Create revision labels	Creates revision labels. Users with this right but not the right to attach labels can still create labels.
Modify revision labels	Changes the properties of revision labels. For example, this right allows a user to freeze labels so that they cannot be adjusted.
Delete revision labels	Deletes revision labels. This action automatically detaches the labels from the folders and items that had those labels. Users with this right but not the right to detach revision labels can still delete revision labels.
Define promotion model	Creates, deletes, and reorders promotion states and edit their properties. After creating a promotion state, you must exit and reenter the Promotion dialog if you want to set access rights for the newly created state.
Create views	Creates views in the current project. This container-level right is available only when you select the View node from the Project Access Rights dialog.
Override default types	Allows users to override the default set of types included when a new view is created.

Related Reference

[Access Rights and Privileges](#)

Folder Access Rights

When you select the **Folder ▶ Advanced ▶ Access Rights** command to display the Folder Access Rights dialog, you see two folder nodes. The rights available from This Folder node apply to the selected folder only. The rights available from the Child Folders node apply to all the child folders of the selected folder. The dialog and following table refer to the current folder. The table describes the access rights that are available from the This Folder node in the Folder Access Rights dialog.

Note: Because This Folder has no Generic Item Container subcategory for access rights, container rights for This Folder are on its Child Folders node. If This Folder is the root folder, these rights are set on the Child Folders node of the View Access Rights dialog.

This access right...	Allows a user or group to...
Generic Object Rights	
See item and its properties	View this folder's Name, Exclude, and Files tabs, which become available when Folder > Properties is selected. The History tab is controlled by the "See folder history" access right. The Link tab is controlled by the "See folder links" access right.
Modify properties	Change folder properties on the folder Name and Exclude tabs. Properties include folder name, description, use of inherited and local exclude lists, and contents of the local exclude list. If the folder is not a root folder, the working folder and alternate working folder settings are also properties. For root folders, the working folders are view properties and not controlled by this access right.
Delete from folder	Delete this folder from its parent folder. Be aware that if you can delete any of this folder's parent folders, you can still delete this folder.
Change item access rights	Change the access rights for this folder. If you change this setting, be sure that you remain one of the users who can change access rights.
See history	See this folder's History tab, which is available when Folder ▶ Properties is selected.
Perform maintenance	Change the revision comments for past revisions.
Set exclusive locks	Lock folders exclusively.
Break exclusive locks	Remove someone else's exclusive lock on the folders.
Label Rights	
Attach/Adjust view labels	Add a view label to this folder. Move a view label from one revision of this folder to another. This right controls direct manipulation of labels for this folder at the folder level. It does not stop users from attaching a view label to this folder when a view label is created.
Detach view labels	Remove a view label from this folder. Be aware that if users can delete view labels, they can detach a view label from this folder by deleting the view label from the view, regardless of the setting for this right.
Attach/Adjust revision labels	Add a revision label to this folder. Move a revision label from one revision of this folder to another. This right controls direct manipulation of revision labels for this folder at the folder level.
Detach revision labels	Remove a revision label from this folder. Be aware that if users can delete revision labels, they can detach a revision label from this folder by deleting the revision label from the view, regardless of the setting for this right.
Link Rights	
See links	See the links involving this folder.
Create links	Link this folder to other folders and items.
Modify links	Change a link for this folder.
Delete links	Delete a link for this folder.

Related Reference

[Access Rights and Privileges](#)

Child Folder Access Rights

When you select the **Child Folders** node from the **Folder Access Rights** dialog box, the available rights apply to the child folders of the selected folder. The **Child Folders** node is also available from the **View Access Rights** dialog box and the **Project Access Rights** dialog box. In these cases, the rights apply to all child folders in the current view or all the child folders in the project, respectively.

The table below describes the access rights available from the **Child Folders** nodes in the **Project Access Rights**, **View Access Rights**, or **Folder Access Rights** dialog boxes.

This access right...	Allows a user or group to...
Generic Item Rights	
See item and its properties	See the selected folder's child folders or the selected project's or view's folders in the folder hierarchy in the left pane on the screen. You can also view the Name and Exclude Properties dialogs, which open when Folder > Properties is selected. The History tab is controlled by the "See folder history" access right.
Modify properties	Change folder properties on the Name and Exclude tabs for child folders. The properties include the folder's name, description, use of inherited and local exclude lists, and the contents of the local exclude list. If a child folder is not a root folder, the working folder and alternate working folder settings are folder properties. If it is the root folder, the working folders are view properties and not controlled by this access right.
Delete from folder	Delete the selected folder's child folders or the selected project's or view's folders from their parent folders. Be aware that if you can delete any of this folder's parent folders, you can still delete this folder.
Change item access rights	Change the access rights for the selected folder's child folders or the selected project's or view's folders. If you change this setting, be sure that you remain one of the users who can change access rights.
See history	See the History tab, which is available when Folder ▶ Properties is selected. This action applies to the selected folder's child folders or the selected project's or view's folders.
Perform maintenance	Change the revision comments for past revisions.
Set exclusive locks	Lock child folders exclusively.
Break exclusive locks	Remove someone else's exclusive lock on the child folders.
Label Rights	
Attach/Adjust view labels	Add a view label to the selected folder's child folders or the selected project's or view's folders. Move a view label from one revision of a child folder to another. This right controls direct manipulation of view labels for child folders at the folder level. It does not stop users from attaching a view label to child folders when a view label is created.
Detach view labels	Remove a view label from the selected folder's child folders or the selected project's or view's folders. Be aware that if users can delete view labels, they can detach a view label from child folders by deleting the view label from the view, regardless of the setting of this right.
Attach/Adjust revision labels	Add a revision label to the selected folder's child folders or the selected project's or view's folders. Move a revision label from one revision of a child folder to another. This right controls direct manipulation of revision labels for child folders at the folder level.
Detach revision labels	Remove a revision label from the selected folder's child folders or the selected project's or view's folders. Be aware that if users can delete revision labels, they can detach a revision label from this folder by deleting the revision label from the view, regardless of the setting of this right.
Link Rights	

See links	See the links involving the selected folder's child folders or the selected project's or view's folders.
Create links	Link the selected folder's child folders or the selected project's or view's folders to other folders and items.
Modify links	Change a link for the selected folder's child folders or the selected project's or view's folders.
Delete links	Delete a link for the selected folder's child folders or the selected project's or view's folders.
Generic Item Container Rights	
Create and place in folder	Create a folder in a parent folder, view, or project in which the Child Folder Access Rights dialog box has this option.
Share/Move out of folder	Share or move a folder in a parent folder, view, or project if its Child Folder Access Rights dialog has this option. Be aware that the access rights set for that folder and its contents, along with any rights set for specific child folders and items within that branch of the folder hierarchy, accompany the folder into the new folder.
Change behavior or configuration	Change the branching ability and configuration of folders that reside in a parent folder, view, or project if its Child Folder Access Rights dialog box has this option.

Related Reference

[Access Rights and Privileges](#)

File Access Rights

When you open the **File Access Rights** dialog box and select the **File** node, the rights apply only to the selected file.

File access rights are also available from the **Folder Access Rights**, **View Access Rights**, and **Project Access Rights** dialog boxes. In these cases, the rights cover all files in the selected folder, view, or project rather than an individual file. Only in these cases are the following container-level access rights available:

- ◆ Add files to a folder
- ◆ Share/move file out of a folder
- ◆ Change file behavior/configuration

The following sections describe the access rights that are available from the **File** nodes in the **Project Access Rights**, **View Access Rights**, **Folder Access Rights**, and **File Access Rights** dialog boxes.

Generic Item Rights

The following section describes generic item rights.

This access right...	Allows a user or group to...
See item and its properties	See files in the files list (upper pane) and view file properties by selecting File Properties .
Modify properties	Change the file properties. Modifiable properties include the archive/file name, description, executable bit setting (useful only for non- Windows platforms), compression, storage options, and custom properties. If used, an alternate property editor (APE) may restrict the properties that can be modified and the users who can modify them still further.
Delete from folder	Delete files from their folders.
Change item access rights	Change access rights for the files. If you change this setting, be sure that you remain one of the users who can change access rights.
See history	See file history in the history pane.
Perform maintenance	Change the revision comments for past revisions.
Set exclusive locks	Lock files exclusively.
Break exclusive locks	Remove someone else's exclusive lock on the files.

Label Rights

The following section describes label rights.

This access right...	Allows a user or group to...
Attach/Adjust view labels	Add a view label to the files. Move a view label from one revision to another. This right controls direct manipulation of view labels for the files at the item level. This right does not stop users from attaching a view label to the files when a view label is created.
Detach view label	Remove a view label from the files. Be aware that if users can delete view labels, they can detach a view label from the files by deleting the view label from the view, regardless of the setting of this right.
Attach/Adjust revision labels	Add a revision label to the files. Move a revision label from one revision to another. This right controls direct manipulation of revision labels for the files at the item level. It can stop users from checking in files with an attached revision label.

Detach revision labels	Remove a revision label from the files. Be aware that if users can delete revision labels, they can detach a revision label from the files by deleting the revision label from the view, regardless of the setting of this right.
------------------------	---

Link Rights

The following section describes link rights.

This access right...	Allows a user or group to...
See links	See the links created for the files.
Create links	Link the files to other items.
Modify links	Change a link to the files.
Delete link	Delete a link that affects the files.

File-Specific Rights

The following section describes file-specific rights.

This access right...	Allows a user or group to...
Check in file	Check in the files.
Check out file	Check out the files.

Generic Item Container Rights

The following section describes item container rights.

This access right...	Allows a user or group to...
Add file to folder	Add files to a folder, view, or project if its File Access Rights dialog box has this option. This right appears only on the File Access Rights dialog boxes associated with a folder, view, or project.
Share/Move out of folder	Share files or move files that reside in a folder, view, or project if its File Access Rights dialog box has this option. This right appears only on the File Access Rights dialog boxes associated with a folder, view, or project. Be aware that the access rights set for any file that is moved or shared go with that file into the new folder.
Change behavior or configuration	Change the branching ability and configuration of files that reside in a folder, view, or project if its File Access Rights dialog box has this option. This right appears only on the File Access Rights dialog boxes associated with a folder, view, or project.

Related Reference

[Access Rights and Privileges](#)

Generic Item Access Rights

The following table describes the access rights that are available from the **File** nodes in the **Project Access Rights**, **View Access Rights**, **Folder Access Rights**, and **File Access Rights** dialog boxes.

This access right...	Allows a user or group to...
See item and its properties	See files in the files list (upper pane) and view file properties by selecting File ▶ Properties .
Modify properties	Change the file properties. Modifiable properties include the archive/file name, description, executable bit setting (useful only for non- Windows platforms), compression, storage options, and custom properties. If used, an alternate property editor (APE) may restrict the properties that can be modified and the users who can modify them still further.
Delete from folder	Delete files from their folders.
Change item access rights	Change access rights for the files. If you change this setting, be sure that you remain one of the users who can change access rights.
See history	See file history in the history pane.
Perform maintenance	Change the revision comments for past revisions.
Set exclusive locks	Lock files exclusively.
Break exclusive locks	Remove someone else's exclusive lock on the files.

Related Reference

[Access Rights and Privileges](#)

Promotion State Access Rights

Each view has its own set of promotion states. Access to these states is controlled by the “Define promotion model” right, which is available from the View node of the Access Rights dialog at the view and project levels. A user with the Define promotion level right can do anything to the promotion model, for example create and delete states, edit their properties, promote a label from one state to another. (Promotion is a subset of editing properties. Anyone who can edit the properties of a state can also promote that state.) and reorder the states within the view.

Access rights that govern access to individual promotion states. These Generic object rights and Promotion state specific rights are available from the Promotion State node of the Access Rights dialog at the view and project levels. They also appear on the access rights for individual promotion states. The rights for an individual promotion state are checked at the state level; if necessary, the checking continues at the view level and eventually the project level. If a user is granted a given right at one level, there is no need to check the next.

When a right is granted at the view level, it applies to all states in the view, unless access is denied at the state level. When a right is granted at the project level, it applies to all the states in all the views within the project, unless access is denied at the state or view levels.

This access right...	Allows a User or Group to...
Change object access rights	Change the access rights for an individual promotion state. If you change this setting, be sure that you remain one of the users who can change access rights. This right is a generic object right. After creating a promotion state, you must exit and reenter the Promotion dialog if you want to set access rights for the newly created state.
Modify label assignment	Change the label assigned to an individual state either by clicking the Promote button or editing the label property. No other properties for the state can be edited unless the user also has the Define promotion model access right from the View node. This right is a promotion state specific right.

Related Reference

[Access Rights and Privileges](#)

Component Access Rights

If you have the server-level access right to “Administer component-level access rights,” you can set component-level access rights from any open component.

The following is a description of the Component Access Rights

This access right...	Allows a user or group to...
Create public filters	Create public filter for this component.
Create public queries	Create public queries for this component.

Related Reference

[Access Rights and Privileges](#)

Component-level Filter Access Rights

The following describes the Filter Access Rights at the Component Level:

This access right	Allows a user or group to...
See object and its properties	See public filters for this component in the filters list (on the toolbar) and view their properties in the Filters dialog.
Modify properties	Change public filter properties for this component. The properties that can be modified for a filter are its list of displayed fields, its sorting and grouping rules, the query associated with it, and its context (the items of the component to which it can be applied).
Delete object	Delete public filters for this component from its list of filters.
Change object access rights	Change access rights for public filters for this component.

Related Reference

[Access Rights and Privileges](#)

Individual Filter Access Rights

The individual filter access rights are described in the table below:

This access right...	Allows a user or group to...
See object and its properties	See the filter in the filters list (on the toolbar) and view its properties in the Filters dialog box.
Modify properties	Change the properties for the filter. The properties that can be modified for the filter are its list of displayed fields, its sorting and grouping rules, the query associated with it, and its context (the items of the component to which it can be applied).
Delete object	Delete the filter from the list of filters
Change object access rights	Change the access rights for the filter.

Related Reference

[Access Rights and Privileges](#)

Component-level Query Access Rights

The following table describes the Query Access Rights at the Component Level:

This access right...	Allows a user or group to...
See object and its properties	See public queries in the Queries dialog and view their properties in the Edit Query dialog.
Modify properties	Change public queries properties for this component. The properties that can be modified are the query's name and its conditions.
Delete object	Delete public queries for this component from its list of queries.
Change object access rights	Change the access rights for public queries for this component.

Related Reference

[Access Rights and Privileges](#)

Individual Query Access Rights

The following table describes the Individual Query Access Rights:

This access right...	Allows a user or group to...
See object and its properties	See this query in the Queries dialog box and view its properties in the Edit Query dialog box.
Modify properties	Change the properties for this query. The properties that can be modified are its name and conditions.
Delete object	Delete this query from the list of queries.
Change object access rights	Change the access rights for this query.

Related Reference

[Access Rights and Privileges](#)

Change Package Access Rights

Change Package item access rights available from the [Project](#) and [View](#) menu [Access Rights](#) menu item.

Generic Change Package Rights

The following section describes generic change package rights.

This access right...	Allows a user or group to...
Create change package	Allows the specified users and groups to create a change package using a View Compare/Merge session..
See change package and its properties	Allows the specified users and groups to view the change package and its properties.
Modify properties	Allows the specified users and groups to modify the change package properties for an uncommitted change package. From the Change Perspective , viewing the selected change package properties is read-only. However, if you select the change package and choose ChangePackage ▸ Open (in VCM) , the change package opens in the VCM session where you can modify its properties.
Delete from view	The properties which can be changed are Allows the specified users and groups to delete a change package from the view. This is an irreversible action.
Change access rights	Allows the specified users and groups to change the access rights for a change package. If you change this setting, be sure that you remain one of the users who can change access rights.
See history	Allows the specified users and groups to see change package history in the History pane.
Set exclusive locks	Allows the specified users and groups to lock change packages exclusively.
Break exclusive locks	Allows the specified users and groups to remove someone else's exclusive lock on change packages.

Label Rights

The following section describes label rights.

This access right...	Allows a user or group to...
Attach/Adjust view labels	Allows the specified users and groups to add a view label to a change package, and move a view label from one revision to another. This right controls direct manipulation of view labels for the change package at the item level. This right does not stop users from attaching a view label to the change package when a view label is created.
Detach view label	Allows the specified users and groups to remove a view label from the change package. Be aware that if users can delete view labels, they can detach a view label from the change package by deleting the view label from the view, regardless of the setting of this right.
Attach/Adjust revision labels	Allows the specified users and groups to add a revision label to a change package and move a revision label from one revision to another. This right controls direct manipulation of revision labels for the change packages at the item level. It can stop users from committing change packages with an attached revision label.

Detach revision labels	Allows the specified users and groups to remove a revision label from a change package. Be aware that if users can delete revision labels, they can detach a revision label from the change package by deleting the revision label from the view, regardless of the setting of this right.
------------------------	--

Related Reference

[Change Package Properties](#)

External Link Access Rights

External Link item access rights available from the **Project** and **View** menu **Access Rights** menu item.

Create external link rights

This access right...	Allows a user or group to...
Create external link	Allows the specified users and groups to create an external link between items in two different views, or on two different servers.
See external link and its properties	Allows the specified users and groups to view the external link and its properties.
Modify properties	Allows the specified users and groups to modify the external link properties. To view the properties of an external link, open both views which contain the link. In one of the views. Click on the Link tab, then right-click the external link and choose Link Properties . You can only modify the description property.
Delete from view	Allows the specified users and groups to delete an external link from the view. This is an irreversible action.

Related Reference

[Change Package Properties](#)

File Compare/Merge Reference

This sections contains reference topics related to File Compare/Merge.

In This Section

[File Compare/Merge Actions](#)

Describes the actions possible in File Compare/Merge.

[File Compare/Merge Keyboard Shortcuts](#)

Presents the keyboard shortcuts for use in the File Compare/Merge .

[File Compare/Merge Options](#)

Describes the option settings for the File Compare/Merge utility.

File Compare/Merge Actions

These are the actions you can perform in File Compare/Merge using menus and toolbar buttons. All of these are available in the standalone File Compare/Merge. Subsets of these are available in the main and embedded File Compare/Merge.

Common Actions

These actions are common to file and folder comparisons and merges.

Action	Description
New File/Folder/Image Comparison	Opens a new file, folder, or image comparison pane.
Undo (CTRL+Z)	Undoes or redoes previous actions.
Redo (CTRL+Y)	
Enable/Disable Independent Scrolling	Controls whether the panes scroll independently or synchronized. If the toolbar button is Enabled (up), as you scroll in one file, the other file automatically scrolls to match the two files. If the toolbar button is Disabled (pressed), the panes use independent file scrolling.
Reload Content (F5)	Reloads the contents of both files. Before reloading the contents, a Save Results dialog box opens giving you the option of saving your results first.
Postponed Opening On/Off	Specifies immediate or delayed file opening. This action is controlled with a toolbar button. With the button in the Postponed Opening Off position (not pressed), as you open a new file, the comparison is immediately performed. With toolbar button in the Postponed Opening On (pressed), the comparison is delayed until this toolbar button turns it on.
Swap Panes	Swaps the left and right panes.
Switch Comparison Modes	<p>Opens the panes in a two-way or three-way horizontal, vertical, or mixed layout. A three-way layout enables you to have a third pane between the two files you're comparing, in which you can merge the two files into a common base version.</p> <p>When you are in a three-way merge layout, you have additional toolbar actions available for automatically merging the two files into a final base version of the file. See Common Parts Auto Merge, Full Auto Merge, and Highlight Changes in this table.</p>
Generate a Report	Opens the Report dialog box where you can select a report name, output type. and report transformer to apply to the report.
Print Comparison Report	Prints the comparison report you generated using Generate Report .
Center/Move Splitter	Enables you to automatically center the splitter bar between the panes using a menu item, or move the splitter with the mouse to resize one of the panes.
Maximize/Minimize Panes (CTRL+M)	<p>Minimizes or maximizes the selected pane.</p> <p>There is also menu items for maximizing/ minimizing/restoring the upper/left or lower/ right panes in a three-way comparison.</p>

File, Folder, and Image Compare/Merge Actions

These actions are specific to file and folder comparisons and merges, and correspond to buttons on the toolbar..

Action	Description
Copy	Copies the selected text or files/folders to the left or right pane, or to the base pane in a three-way comparison. Copied files or folders are marked with a check mark, and you are prompted to commit the changes.

	<p>There are specific toolbar buttons for doing this, and when you hover the mouse cursor over a highlighted area, a Copy button also display in the pane in the highlighted area.</p>
Delete	<p>Deletes the highlighted text or files/folders from the left, right, or base pane. Deleted files or folders are marked with a circle icon containing an X, and you are prompted to commit the changes.</p> <p>There are specific toolbar buttons for doing this, and when you hover the mouse cursor over a highlighted area, a Delete button also display in the pane.</p>
Move Image	<p>Moves the image by clicking and dragging on the image when the Move Image toolbar button is pressed.</p>
Append Text (ALT+SHIFT+RIGHT)	<p>Appends the selected text from one pane the other pane or panes. The button for appending text appears when the mouse cursor hovers over a highlighted area.</p> <p>When you hover the mouse cursor over a highlighted area, an Append button also display highlighted area of the pane.</p>
Insert/Replace Text (ALT+RIGHT)	<p>Inserts or replaces the selected text from one pane the other pane or panes.</p> <p>When you hover the mouse cursor over a highlighted area, an Insert or Replace button also display highlighted area of the pane.</p>
Find Specified Text (CTRL+F)	<p>Searches for a text string or regular expression in a text-based file. You can choose whether to use case sensitivity and whole words only.</p>
Previous Change (F7or ALT+UP) Next Change (F8or ALT+UP)	<p>Quickly navigates to the next or previous change in the selected pane. In addition to menu commands, there are toolbar buttons, and buttons at the bottom of each pane in the scroll bar that navigate your changes.</p>
Go To Line (CTRL+G)	<p>Opens the Go To Line dialog box where you can specify the line number to go to.</p>
Inline Differences	<p>Displays or hides inline differences when the toolbar button is pressed. You can customize how inline differences are displayed in the File Compare/Merge Options dialog box available on the Tools menu.</p>
Split Differences	<p>Splits blocks of changed lines to represent inserted or deleted blocks (by default, signified by a light green color).</p>
Highlight Changes	<p>Used in conjunction with a three-way comparison, it prevents File Compare/Merge from automatically recalculating differences on any file modifications, and helps you see which blocks were inserted or deleted to or from the base file.</p>
Nonconflicting Parts Auto Merge	<p>Used in conjunction with a three-way comparison,File Compare/Merge places only the nonconflicting portions of the two compared files into the base file.</p>
Full Auto Merge	<p>Used in conjunction with a three-way comparison, File Compare/Merge parses the two files from top to bottom, examines the changes and automatically decides whether to keep or add each change to the final base version of the file.</p>
Encoding	<p>Enables you to specify a desired character encoding (codepage) for the displayed files using the Charset Selection dialog box.</p> <p>File Compare/Merge displays encoding information in the bottom-right corner of the Status Bar. You can double-click the encoding icon to open the Charset Selection dialog box.</p>
Save Changes to Disk	<p>Saves changes to your folders to disk. When you make a change to folders, the changed folder or file is marked with a check mark, an you are prompted to commit the change.</p>

Auto Commit	Specifies immediate or deferred committing of changes to disk.
Enable/Disable Synchronized Selection	Enables automatic selection of the same file in both panes, or all panes in a three-way comparison. This is enabled by default. If turned off using the toolbar, only the file in the pane with focus is selected.
Expand/Collapse All Nodes In Tree	Expands or collapses all the nodes in the tree. These actions are available on the toolbar.
Expand Only Tree Nodes Containing Changes	Expands only tree nodes that contain changes in the folders. These actions are available on the toolbar.
Extend/Compact View	Toggles between a compact and extended view of your folder comparisons. The compact view collapses empty color blocks into single lines. The default is a compacted view.
Smart Tree Diff	Performs a smart comparison on the folders, highlighting differences such as renamed and modified components or cross-linked components.
Show/Cut Plain Differences	<p>Enables you to see differences that conform to folder structure and file names simultaneously, with differences found by the smart comparison feature (Smart Tree Diff).</p> <p>This toolbar button is only available if you have enabled the smart comparison feature by pressing Smart Tree Diff on the toolbar.</p> <p>The Show/Cut Plain Differences button never displays on the toolbar if you have chosen to use the Alternative Tree Diff Algorithm on the Smart Comparison page for folder comparisons in the File Compare/Merge Options.</p>
Zoom	Zooms in or out on the images being compared when clicking the plus or minus toolbar buttons.
Drag and Drop	Allows you to drag and drop content from one pane to the other. You can move or copy the content depending on whether you use the CTRL key when dragging.

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Comparing a Local File with a Repository File](#)
[Comparing Historical File Contents](#)
[Comparing Two Local Files](#)
[Editing Files in a File Compare/Merge Session](#)
[Saving Files Modified in a File Compare/Merge Session](#)
[Comparing Folders](#)
[Comparing Images](#)
[Merging a Local File with the Tip Revision](#)
[Merging Two Local Files](#)
[Merging Folders](#)
[Generating Reports from a File Compare/Merge Session](#)
[Customizing Compare and Merge Reports](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Keyboard Shortcuts](#)

File Compare/Merge Keyboard Shortcuts

This table presents the keyboard shortcuts for use in File Compare/Merge.

Action	Keyboard Shortcut
New File Comparison	CTRL+N
New Folder Comparison	CTRL+SHIFT+N
New Folder Merge	CTRL+SHIFT+M
New Image Comparison	CTRL+ALT+N
Save All	CTRL+SHIFT+S
Print	CTRL+P
Undo	CTRL+Z
Redo	CTRL+Y
Cut	CTRL+X
Copy	CTRL+C
Paste	CTRL+V
Find	CTRL+F
Find Next	F3
Find Previous	SHIFT+F3
Go To Line	CTRL+G
Previous Change	F7 OR ALT+UP
Next Change	F8 OR ALT+DOWN
Next Tab	F6
Previous Tab	SHIFT+F6
Maximize/Minimize Pane	CTRL+M
Center Splitter	CTRL+H
Move Splitter	CTRL+SHIFT+S
Reload	F5
Open Options Dialog Box	CTRL+O
Replace Text Left To Right (2-way merge)	ALT+RIGHT
Replace Text Right To Left (2-way merge)	ALT+LEFT
Append Text Left To Right (2-way merge)	ALT+SHIFT+RIGHT
Append Text Right To Left (2-way merge)	ALT+SHIFT+LEFT
Delete Text in Left Pane (2-way merge)	ALT+CTRL+LEFT
Delete Text in Right Pane (2-way merge)	ALT+CTRL+RIGHT
Move Text Up	ALT+SHIFT+UP
Move Text Down	ALT+SHIFT+DOWN
Replace Text Center To Left	ALT+<
Replace Text Center To Right	ALT+>

Replace Text Left To Center	ALT+M
Replace Text Right To Center	ALT+/
Append Text Left To Center	ALT+SHIFT+M
Append Text Right To Center	ALT+SHIFT+/
Append Text Center To Left	ALT+SHIFT+<
Append Text Center To Right	ALT+SHIFT+>

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Actions](#)

File Compare/Merge Options

Option settings globally impact product behavior. Many options are provided to enable you to configure File Compare/Merge to function in a way that you desire. Option settings are provided at a general level and for both file and folder comparisons at a more granular level.

The File Compare/Merge Options are available using **Tools ▸ Options** on the main menu of the File Compare/Merge window, or pressing **CTRL+SHIFT+P**. A tree hierarchy on the left side of the File Compare/Merge Options dialog box divides the File Compare/Merge options into three main categories, **General**, **File Comparison**, and **Folder Comparison**. Clicking on a node in the hierarchy displays its corresponding page of option settings in the right side of the dialog box.

You can reset the options to their default values by deleting the `config.properties` user settings file found in your `<user home>\.fcm` folder. (For example, in Windows it would be `C:\Documents and settings\<user home>\.fcm\config.properties`).

In This Section

[General \(File Compare/Merge Options\)](#)

Describes General Comparison Options in the File Compare/Merge Options dialog box.

[File Comparison \(File Compare/Merge Options\)](#)

Describes Filer Comparison Options in the File Compare/Merge Options dialog box.

[Folder Comparison \(File Compare/Merge Options\)](#)

Describes Folder Comparison Options in the File Compare/Merge Options dialog box.

General (File Compare/Merge Options)

These are general options for File Compare/Merge. To access these options, choose **Tools ▸ Options ▸ General**.

Option	Description
Open Panes For:	<p>Opens File Compare/Merge ready for making the selected type of comparison on start-up: A File Comparison, A Folder Comparison, or the Last Type of Comparison made.</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
File History Limit	<p>Populates the file history list with a maximum value. You can view the file history by clicking the DOWN ARROW to the right of Open New Comparison on the toolbar.</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
Immediately Start Comparison As New File/Folder Is Entered	<p>Starts a comparison as soon as the file/folder is opened. If this option is not set, open the file/folder to compare, and click Reload Content on the toolbar to begin the comparison.</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
Open Comparison In Read-only Mode	<p>Opens files that you are comparing in read-only mode.</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
Allow Editing For Read-only Files/Folders	<p>Allows you to edit read-only files/folders.</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
Darken Pane While In Read-only Mode	<p>Darkens the UI pane to indicate the file/folder is read-only,</p>
Enable Quick Start	<p>Places the File Compare/Merge Quick Start icon in your Windows System Tray. (Note: This could be invisible if File Compare/Merge cannot find the required .dll file.)</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
Allow Only One Running Instance	<p>Allows starting of only one instance of File Compare/Merge .</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
3-way Merge: Start With Empty Center Pane	<p>Opens an empty third pane on switching to a 3-way merge..</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
3-way Merge: Allow Editing Only In Center Pane	<p>Allows you to only edit the center pane (merged results) on switching to a 3-way merge.</p> <p>This option is not available when launching File Compare/Merge from within the client.</p>
3-way Merge: Put Left Scroll Bar on Center Pane	<p>Adds a left scroll bar to the center pane for a 3-way merge.</p>

Tabs: On Bottom	Locates tabs containing your file/folder comparisons at the bottom of the UI by default. If this option is off, the tabs are located at the top of the UI just beneath the toolbar. This option is not available when launching File Compare/Merge from within the client.
Tabs: Start New Comparison In A New Tab	Starts new comparisons in a new tab by default. This option is not available when launching File Compare/Merge from within the client.
Tabs: Show Only One Comparison	Only allows one file/folder comparison. Selecting this option disables the Start New Comparison in a New Tab option. This option is not available when launching File Compare/Merge from within the client.
For Next/Previous Change: Always Adjust Scroll Bar	Automatically adjusts the scroll bar to the location of the next or previous change.
For Next/Previous Change: Adjust Scroll Bar Only If The Difference Is Off Screen	Automatically adjusts the scroll bar to the location of the difference if it is not visible on the screen.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)

[Merging a Local File with the Tip Revision](#)

[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)

[File Compare/Merge Keyboard Shortcuts](#)

[File Compare/Merge Actions](#)

File Comparison (File Compare/Merge Options)

These option apply only to file comparisons. To access these options, choose **Tools ▶ Options ▶ File Comparison**.

Main File Comparison Options

Below are the options available on the first page of the File Comparison node.

Item	Description
Ignore All White Space	ASCII characters 0x01 to 0x20 are removed from the text.
Ignore Leading and Trailing White Space	Ignores all leading or trailing white space in a line.
Ignore Differences in End-Of-Line Separators (CR and LF)	Ignores differences in end-of-line CR LF separators when comparing files. There are Windows (CR LF), Unix (LF) and Mac (CR) line endings that can affect your file comparisons. This is especially useful if you are comparing files that have been generated on a variety of operating systems.
Ignore Case	Ignores differences in character case within lines.
Insert Spaces Instead of Tabs	Inserts spaces instead of tabs.
Tab Size	Specifies the tab size.
Show Invisible Characters	Shows invisible characters in the editor panes, such as the trailing carriage return and/or line feed (CR LF) characters, and tabs and spaces. This option does not affect the end-of-line characters used when files are saved -it only displays them visually in the editor pane. You can change the line endings using the Change Line Separator On Load, Use Line Separator On Enter, Change Line Separator On Copy, Change Line Separator On Paste, and Change Line Separator On Save Options .
Change Line Separator When File Opens	Designates line separators. You can choose from Platform Specific, No Change, Windows, Unix, or Mac .
Insert Line Separator On Enter	Determines the line separator to use whenever you enter a new line in the view pane. You can choose from Auto (need reload), Platform Specific, Windows, Unix, or Mac .
Change Line Separator On Copy	Determines the line separator that will be used in the copied text whenever you copy text in the view pane. You can choose from Platform Specific, No Change, Windows, Unix, or Mac .
Change Line Separator On Paste	This setting determines the line separator that will be used in pasted text. Using the combo box, you can choose from Auto (need reload), Platform specific, No change, Windows, Unix, or Mac.
Change Line Separator On Save	Determines the line separator that will be used when saving changes. You can choose from Platform Specific, No Change, Windows, Unix, or Mac .
Editor Font	Changes the View pane editor font and size.
Show Line Numbers	Displays line numbers on the left side of each of the editor panes.

When File Opened, Auto Scroll To Difference	Scrolls automatically to the first difference found in your file comparison.
When File Opened, Reset Scroll to Beginning	Scrolls automatically to the beginning of the file in your file comparison.
Default Encoding	Specifies an encoding value to apply to your files. The Default Encoding value is System-Dependent , and it is applied to any new files that you open in File Compare/Merge.
Change Encoding For Opened Files	Changes the encoding automatically for files that you currently have loaded in File Compare/Merge. The value that you select for the encoding in the Default Encoding option is applied.
Inherit Encoding From Opened File	Specifies that if you replace the current file in a file comparison, the file that you open automatically inherits the encoding of the file that it is replacing. Without this setting activated, the file gets its encoding from the Default Encoding option.
Preliminary Comparison:	Internally compare two files (or two versions of the same file) first to decide if they are the same. If they are the same, then prompts the user that "Files are identical. Continue opening diff?". These options let you use a different set of settings when doing this initial comparison, than when doing the comparison which is actually displayed.
Preliminary Comparison: No Comparison	Does not perform an initial comparison, and never displays a prompt. In this case, if you compare two identical files, FCM just launches and shows the files.
Preliminary Comparison: Do Comparison Ignoring All Settings	Performs a comparison which ignores the settings.
Preliminary Comparison: Do Comparison Using All Settings	Performs a comparison which uses all the settings.
Preliminary Comparison: Do Comparison Firstly	First performs a comparison ignoring the settings. If this comparison indicates there is a difference, the FCM performs another comparison using the settings. If the second comparison says there is a difference, then it prompts the user. The intent of this last option is to prompt if either type of comparison (Ignoring All Settings and Using All

Inline Differences

Use this options to control how inline differences are handled.

Option	Description
Show Inline differences	Performs detailed inline comparisons on changed lines highlighting insertions, changes, and so on.
Compare Lines By Characters As A Block	Compares and highlights differences inside a sequence of changed lines. All characters in the lines are scanned, and the text is divided by lines. You must enable Show Inline Differences to activate this option.
Compare Lines By Characters	Compares the characters in each line with the characters in the opposite line. For example, characters in the first line of the sequence are compared with the characters in the first line of the opposite sequence, characters in the second line of the sequence are compared with the characters in the second line of the opposite sequence and so on. This option is automatically activated when you enable Show Inline Differences .
Compare Lines By Words	Splits lines into words (or lexemes) using standard separators, and compares them. You must enable Show Inline Differences to activate this option.
Separate Differences	Enables you to Compare Lines By Characters , Compare Lines By Words , or Compare Lines By Words - Best Choice . The default setting is to compare lines at a granular level, by characters. Normally, File Compare/Merge presents sequences of differences for changed lines as a block. In other words, you can perform operations, such as Replace , Append , or Delete only on the block. If you have enabled this option, File Compare/Merge does not combine the sequences in a block. Instead it considers each pair of changed lines as a block. You must enable Show Inline Differences to activate this option.
Best Choice	Compares all pairs of lines and selects the best ones. You must enable Show Inline Differences to activate this option.
Minimum Word Size	Represents the minimum number of characters between separators that is considered as a word. You must enable Show Inline Differences to activate this option.
Acceptance Weight (Words*100+Chars)	Provides the criteria for File Compare/Merge to decide which lines should be considered a match and have their differences highlighted. The acceptance weight is equal to the number of hit words multiplied by 100, plus the total number of equal (matched) characters in the comparing lines. You must enable Show Inline Differences to activate this option.
Color Background	Highlights the background of the different characters in the line. You must enable Show Inline Differences to activate this option. The Highlight Characters option is not available with this option selected.
Strikethrough	Crosses out the characters in the line to show differences. You can also activate the Highlight Characters option along with this option.

Strikethrough And Highlight	Crosses out and highlights the characters in the line to show differences.
Underscore	Underscores the characters in the line to show differences. You can also activate the Highlight Characters option along with this option.
Underscore and Highlight	Underscores and highlights the characters in the line to show differences.
Left Color/Right Color	Changes the color within the line. You can set colors for both the right and left editor panes.

Smart Comparison


You can apply these settings to compare files at a more granular level.

Option	Description
Advanced Search Of Differences	Uses a special algorithm to find swapped blocks and perform efficiently on large files, enabling you to use this option to improve performance when comparing large files. This option must be selected to enable the Detect swapped blocks option.
Detect Swapped Blocks	Detects swapped blocks of text. The Advanced Search Of Differences option must be selected to enable the Detect swapped blocks option, Hide Links Between Unselected Blocks , Shorten Differences , and Include Only Blocks With Unique Lines options are automatically selected by default with this option activated.
Hide Actual Difference Links Between Selected Blocks	Hides actual difference links between selected blocks. You can optionally activate this option when you activate Detect Swapped Blocks .
Hide Links Between Unselected Blocks	Hides links between unselected blocks. This option is activated by default if you choose to Detect Swapped Blocks .
Shorten Differences	Prevents Detect Swapped Blocks from creating expanded blocks (appending identical lines to blocks of changed lines).
Include Repeated Blocks	Includes repeated blocks. You can optionally activate this option when you activate Detect Swapped Blocks .
Ignore Repeated Blocks	Ignores repeated blocks. You can optionally activate this option when you activate Detect Swapped Blocks .
Include Only Blocks With Unique Lines	Ignores blocks not containing unique lines when a comparison is performed if Detect Swapped Blocks is activated.

Patterns

Use this option page to edit a list of patterns (or regular expressions) that are used in file comparisons. These options are useful for filtering out unimportant changes that you do not necessarily want to see in the editor panes.

Patterns conform to the standard syntax of Regular Expressions. For examples, see <http://www.regular-expressions.info/>.

Option	Description										
Ignore	Allows you to ignore specified patterns during a file comparison. To ignore patterns, check this option, and then choose one or more options from the pattern list to ignore. You can also create new patterns using New to open the Ignorable Pattern dialog box where you can enter regular expression syntax to search for sequences of characters.										
Allow Gluing of Blocks Separated By Pattern-like Lines	<p>Allows you choose to glue lines that are empty, or that contain only braces. You can also modify the existing patterns, or add your own patterns.</p> <p>For example, if you have three lines of differences that you want to move from <code>file1.txt</code> to <code>file2.txt</code>, and those different lines are separated only by blank lines or lines with braces in them, without this option, you would have to resolve each line of difference separately.</p> <p>When Allow gluing of blocks separated by pattern-like lines is enabled AND a pattern (custom or predefined) is enabled, then AND ONLY THEN, the "Glue delimiter lines" button  is displayed in the FCM toolbar. Selecting this toolbar button merges the three lines of differences into a single difference, because the lines between the differences match the glue pattern of "empty line" specified in the FCM Pattern options. Now, it only takes one click of a button to move all three lines of differences from one file to the other.</p>										
New/Change/Remove	Enables you to add new patterns, change existing patterns, or remove existing patterns. New and Change display the Patterns dialog box where you define the pattern.										
Ignorable Pattern/Glue Pattern dialog boxes	<p>Uses New and Change to display the a dialog box where you define a pattern.</p> <p>In the Glue Pattern dialog box , you also have an option to add file extension(s) to apply the glue pattern to. If you are applying the glue pattern to more than one file type, separate them with commas. For example: <code>java, cpp, c, h, hpp</code>.</p> <table border="1"> <tr> <td>Name</td><td>Specifies the name for the pattern. The name displays in the corresponding pattern/ glue list exactly as you enter it in the dialog.</td></tr> <tr> <td>Pattern</td><td>Specifies the pattern (expression). You can use regular expression syntax. For example, <code>^[\t]*//.*\$</code> or <code>[{}\s]*</code>.</td></tr> <tr> <td>Left Text Box</td><td>Provides a place to type sample test text.</td></tr> <tr> <td>Right Text Box</td><td>Displays the results after performing a pattern test.</td></tr> <tr> <td>Test</td><td>Applies the pattern to the sample test text. The results display in the text box on the right of the dialog.</td></tr> </table>	Name	Specifies the name for the pattern. The name displays in the corresponding pattern/ glue list exactly as you enter it in the dialog.	Pattern	Specifies the pattern (expression). You can use regular expression syntax. For example, <code>^[\t]*//.*\$</code> or <code>[{}\s]*</code> .	Left Text Box	Provides a place to type sample test text.	Right Text Box	Displays the results after performing a pattern test.	Test	Applies the pattern to the sample test text. The results display in the text box on the right of the dialog.
Name	Specifies the name for the pattern. The name displays in the corresponding pattern/ glue list exactly as you enter it in the dialog.										
Pattern	Specifies the pattern (expression). You can use regular expression syntax. For example, <code>^[\t]*//.*\$</code> or <code>[{}\s]*</code> .										
Left Text Box	Provides a place to type sample test text.										
Right Text Box	Displays the results after performing a pattern test.										
Test	Applies the pattern to the sample test text. The results display in the text box on the right of the dialog.										

Report Transformers

The **Report Transformer** options page enables you to define new transformers, edit existing transformers, or remove transformers from the list. These options are available whenever you choose to generate a report (click

Generate Report in the toolbar). You can choose to use an empty transformer, use one of the predefined transformers, or create your own.

Option	Description
Report Transformers List	Displays the list of report transformers available for use in transforming XML files to various outputs.
Add/View & Edit	Allows you to add, view, or edit a report transformer XSL file using the Report Transformer Configuration dialog box.
Remove	Allows you to remove a selected report transformer from the list.

Color Preferences

Use the **Color Preferences** to control the colors displayed for file comparisons in the editor panes. Changes to the colors apply to each of the editor panes.

Option	Description
Text Type	<p>Assign distinct background colors to aid in identifying differences and similarities in files. This option displays a list of the types of text that can be color coded in the File Compare/Merge window. The Background Color button automatically changes colors to reflect the type of text selected. The text preview area shows the current color configuration settings. You can change the background color for the following types of text:</p> <ul style="list-style-type: none">– Unchanged text line– Changed text line– Inserted text line– Deleted text line– Conflict text line– Transposed text line– Transposed and changed text line– Transposed inserted text line– Frozen text line. Note that this is also referred to as Made change (Advanced merge mode) in the Text type drop down list . This type of text represents lines in the non-result panes that have been merged.
Background Color	Displays the color of the selected Text Type . This button also launches the Select Color dialog box where you can specify a different color.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)

[Comparing a Local File with a Repository File](#)

[Merging a Local File with the Tip Revision](#)

[Comparing Historical File Contents](#)

[Comparing Two Local Files](#)

[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)

[File Compare/Merge Keyboard Shortcuts](#)

[File Compare/Merge Actions](#)

Folder Comparison (File Compare/Merge Options)

These option settings apply only to folder comparisons which are only available in the external version of File Compare/Merge, accessed from the Start Menu. To access these options, choose **Tools** ► **Options** ► **Folder Comparison**.

Main Folder Comparison Options

In a single paragraph, describe the major components in the diagram.

Option	Description
Tree Font	Sets the font style and size displayed in the result panes.
Ignore Extensions	Designates any file extensions to ignore while comparing folders, for example, <code>dll;exe</code>
Show Only Extensions	Designates only specific file extensions to show while comparing folders.
Ignore Folders	Enables you to list folders that you wish to ignore while comparing folders.
Do Not Show Empty Folders	Ignores empty folders while comparing folders.
Use Compact View	Uses a compacted view if you have files in one folder that do not exist in another folder (files to be inserted). The view showing the folder without those files displays a collapsed color block as a line indicating that the files are not in the other folder. If the option is not selected, the view displays a full color block representing the missing files for the folder that does not contain the files.
Highlight Uncommitted Changes	Highlights uncommitted changes in the View pane.
Hide Unmodified Components	Hides all folders in which there are no differences so the only folders visible in the tree are the ones with differences.

Smart Comparison

You can apply these settings to compare folders at a more granular level. File Compare/Merge uses these options to find similar files regardless of folder structure and folder names.

Option	Description
Use File Names In File Matching	Specifies whether to use the file name or the content for comparison. If true, the file name is used to identify files that are the same. If false, the content will be used to identify files which are the same.
Match Files By Name When No Content Match Is Found	If true, File Compare/Merge uses the file name to compare file contents when the application cannot find any matches in the contents.
Use Folder Names In Folder Matching	Specifies whether to use the folder name for comparison. If true, the folder name is used to identify folders that are the same. If false, File Compare/Merge matches folders regardless of the name.
Maximum Number Of Lines In Matching Files	Specifies the number of lines read when comparing the contents of two files. Differences past the specified maximum number of lines are ignored
Rate Of Proximity In File Matching	The percentage of similarity in comparing files with similar content.
Use Alternative Tree Diff Algorithm	This algorithm behaves as follows: The original folder structure for matched files is not kept and you cannot see (by using the quick switch button on

the toolbar) the conventional comparison results and make appropriate copy operations.

Only the first encountered file that conforms to the matching criterion is kept. Uses a different approach to match folders that contain similar files.

Report Transformers

The **Report Transformer** options page enables you to define new transformers, edit existing transformers, or remove transformers from the list. These options are available whenever you choose to generate a report (click **Generate Report** in the toolbar). You can choose to use an empty transformer, use one of the predefined transformers, or create your own.

Option	Description
Report Transformers List	Displays the list of report transformers available for use in transforming XML files to various outputs.
Add/View & Edit	Allows you to add, view, or edit a report transformer XSL file using the Report Transformer Configuration dialog box.
Remove	Allows you to remove a selected report transformer from the list.

Color Preferences

Use the **Color Preferences** to control the colors displayed for file comparisons in the editor panes.

Option	Description
Component Type	Displays a list of the types of components that can be color coded in the File Compare/Merge window. The Background Color button automatically changes colors to reflect the type of component selected. The text preview area shows the current color configuration settings.
Background Color	Displays the color of the selected Component Type . This button also launches the Select Color dialog box where you can specify a different color.

Related Concepts

[Overview of File Compare/Merge](#)

Related Procedures

[Comparing and Merging Files](#)

[Comparing Folders](#)

[Merging Folders](#)

[Generating Reports from a File Compare/Merge Session](#)

[Customizing Compare and Merge Reports](#)

[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)

[File Compare/Merge Keyboard Shortcuts](#)

[File Compare/Merge Actions](#)

View Compare/Merge

This sections contains reference topics related to View Compare/Merge in StarTeam.

View Compare/Merge is only available in the StarTeam Cross-Platform Client, and the VCMUtility.

In This Section

[View Compare/Merge Wizard](#)

Describe the View Compare/Merge Wizard steps

[View Compare/Merge Session Filters](#)

[Change Package Properties](#)

Lists and defines the change package properties.

[Versioning Object Properties](#)

Lists and defines the common properties of all versioning properties for objects that use versioning.

[View Compare/Merge Terminology](#)

Glossary of View Compare/Merge terms.

View Compare/Merge Wizard

This section describes the individual pages of the **View Compare/Merge Wizard**. The order in which the pages occur in the wizard depends on the type of comparison you are performing and from which view you start the wizard.

Each step of the **View Compare/Merge Wizard** contains a button at the bottom right the page which, when clicked, opens an information pane that shows you the details of the choices you have made for the VCM session up to that step in the **View Compare/Merge Wizard**. To hide this information pane, click the button again and it will disappear. The advantage of this information pane is that you can review in one place all the details of the session before you click **Finish** without having to go back and forth through the different pages of the wizard to see what you selected. That way, if you decide you want to make a different choice, you can go back to a specific page of the wizard, make any changes, then click **Finish** at that point.

In This Section

[View Compare/Merge Wizard: Select Compare/Merge Type](#)

[View Compare/Merge Wizard: Include Items](#)

[View Compare/Merge Wizard: Include Selected Items](#)

[View Compare/Merge Wizard: Select Source View](#)

[View Compare/Merge Wizard: Select Target View](#)

[View Compare/Merge Wizard: Select Source Configuration](#)

[View Compare/Merge Wizard: Select Target Configuration](#)

[View Compare/Merge Wizard: Set Options](#)

[View Compare/Merge Wizard: Exclude Properties](#)

View Compare/Merge Wizard: Select Compare/Merge Type

Use the Welcome page of the **View Compare/Merge Wizard** to select the type of compare/merge to perform.

Item	Description
Select The Type Of Compare Or Merge	<p>Specifies the type of compare merge to perform:</p> <p>Compare only: Compare changes between one related view and another, or between two different configurations of the same view. (Default)</p> <p>Promote: Compare or merge changes from a child view to its parent.</p> <p>Rebase: Compare or merge changes from a parent to its child view.</p> <p>Replicate: Compare or merge changes from one related view to another.</p>
Use Current View (name) For:	<p>Specifies whether the current view should be used as the source view or the target view, or as both the source and target (compare only). If you choose Promote or Rebase as the merge type, the wizard may automatically select the appropriate option based on whether the parent view or a child view is the current view.</p>

View Compare/Merge Wizard: Include Items

Use this page to limit the compare/merge to specific types of items. This step only appears if you start the **View Compare/Merge Wizard** from the [View](#) menu.

Item	Description
Revision Label	Compares items with the selected revision label only. When checked, the drop-down box is enabled and displays the available revision labels.
Include	Specifies which types of items in the view to include in the compare/merge.

View Compare/Merge Wizard: Include Selected Items

Use this page to limit the compare/merge to specific types of items. This step of the wizard only appears if you right-click specific items on the component tabs in the upper pane choose [Advanced View Compare/Merge](#) on the context menu to start the compare/merge session.

Item	Description
Include	<p>Specifies which types of items to include in the compare/merge.</p> <p>The Include tree displays an enabled check box for each item type you have selected, and also gives you the option of selecting all files linked to the selected item type if it is a process item. When you check Selected Folders in the Include tree, you can also check Subfolders to include all contained child folders and their children in the compare.</p> <p>Selecting a folder from the Folder component tab instead of the in the Folder Tree enables you to click other item tabs and select additional items from other folders to include in the compare/merge.</p>

View Compare/Merge Wizard: Select Source View

Use this page of the wizard to select the source view for the compare/merge.

Item	Description
View Tree	Uses the selected view as the source view for the compare/merge. Click to select.
View Name	Displays the name of the view selected in the View Tree .
View Description	Displays the description of the selected view. A view description is entered in the Description field of the View Properties dialog box.

View Compare/Merge Wizard: Select Target View

Use this page of the wizard to select the target view for the compare/merge.

Item	Description
View Tree	Uses the selected view as the target view for the compare/merge. Click to select.
View Name	Displays the name of the view selected in the View Tree .
View Description	Displays the description of the selected view. A view description is entered in the Description field of the View Properties dialog box.

View Compare/Merge Wizard: Select Source Configuration

Use this page of the **View Compare/Merge Wizard** to restrict the source view to a specific configuration in the View Compare/Merge session.

Item	Description
Source View Configuration for (view name)	<p>Specifies a configuration to use for the View Compare/Merge session so you can isolate the view compare/merge from changes in the current view. Select from:</p> <p>Current Configuration: This uses the view's current configuration.</p> <p>Labeled Configuration: This uses the configuration with the revision label you specify.</p> <p>Promotion State Configuration: This uses the configuration with the selected promotion state.</p> <p>Configuration As Of: This uses the configuration from the specified date and time.</p>

View Compare/Merge Wizard: Select Target Configuration

Use this page of the **View Compare/Merge Wizard** to restrict the target view to a specific configuration in the View Compare/Merge session.

Item	Description
Target View Configuration for (view name)	<p>Specifies a configuration to use for the View Compare/Merge session so you can isolate the view compare/merge from changes in the current view. Select from:</p> <p>Current Configuration: This uses the view's current configuration.</p> <p>Labeled Configuration: This uses the configuration with the revision label you specify.</p> <p>Promotion State Configuration: This uses the configuration with the selected promotion state.</p> <p>Configuration As Of: This uses the configuration from the specified date and time.</p>

View Compare/Merge Wizard: Set Options

This page of the **View Compare/Merge Wizard** contains options that will control the beginning of the View Compare/Merge session. You can change individual default actions in either the **Compare Perspective** or the **Merge Perspective** during the View Compare/Merge session.

Option	Description
Auto-merge Properties	View Compare/Merge automatically merges the view, folder, or item properties at the beginning of the View Compare/Merge session. For example, if a property has changed only in the source, the value of that property in the target is changed to match. (Default)
Auto-merge Files	View Compare/Merge automatically merges files without conflicting differences at the beginning of the View Compare/Merge session. (Default)
Match Files By Name	View Compare/Merge matches files with the same file names, even if the object IDs are different, or if the files cannot otherwise be matched. For example, if the same line has changed in both a source and target text file, View Compare/Merge cannot automatically merge the two lines, but if the line has changed only in the source, those changes can also be made to that line in the target. When you match source and target files by name, and the object IDs are different, no branch point can be used to locate a common ancestor when merging or comparing the two files. However, if you have compared these two files (in this source-target direction) before, there may be merge points.
Treat File Names As Case Sensitive	If Match Files By Name is checked, file matching is case sensitive.
Ignore Merge Points	If checked, View Compare/Merge ignores existing merge points by default. If unchecked, View Compare/Merge uses the last recorded merge point between two items to determine if either or both items has changed. When View Compare/Merge performs a merge on a text file, the revision of one of the files (usually the source file) is used as the least common ancestor to perform the merge. When merge point usage is not enabled, the least common ancestor is determined by finding the last branch point common to the two files. It should be very rare that you would ever want to ignore merge points.
Fix/Find Floating Child Shares	<p>Fixes any shares that are floating by pinning them in a Rebase or Promote, since View Compare/Merge cannot merge floating shares. In a Compare-only session, VCM simply finds the floating child shares for you.</p> <p>In Rebase and Replicate merge operations, this option specifies whether each target view item found that is a floating share of a source view item should be “fixed” by pinning it. When a target view item is a floating child share of a source item (which implies that the target item has not branched), differences will not be detected between the source and target item during VCM sessions because changes to the source item immediately float to the child item. View Compare/Merge best practices suggests that child shares should always be pinned, allowing changes to propagate from the source to target view in a controlled manner. This option allows floating child items found by View Compare/Merge to be “fixed” by pinning them to the parent item revision. Specifying this option has a performance cost due to the extra commands required to check each target item examined during the compare phase. Tip: You might want to check this the first time you perform a compare/merge on a view, then uncheck it for successive comparisons to speed up your sessions."</p>
Break Locks Automatically	Attempts automatically to break an existing lock if the item is already locked by another user, and if the item needs to be locked due to one of

the locking options below being set. Breaking locks requires special permissions. **Break Locks Automatically** is only relevant when View Compare/Merge wants to lock an item. And it only locks items for actions that make changes. For example, if you are in the middle of a session, and you change an action from **Ignore** to **Delete**, View Compare/Merge may break someone else's lock on the item that is going to be deleted at commit time.

Lock Source For Difference	Locks source items for differences that will result in a change. Some actions, such as Ignore , will not result in a change, and therefore do not require a lock. Locking usually happens at compare time, but may also happen later in the session if the action for a difference is overridden. VCM will not lock reverse-shared items in the source view unless you check this option in the View Compare/Merge wizard.
Lock Target For Difference	Locks target items for differences that will result in a change. Some actions, such as Ignore , will not result in a change, and therefore do not require a lock. Locking usually happens at compare time, but may also happen later in the session if the action for a difference is overridden. (Default)
Start With Merge Perspective Current	Starts the View Compare/Merge session in the Merge Perspective . If unchecked, View Compare/Merge starts in the Compare Perspective . (Default)

View Compare/Merge Wizard: Exclude Properties

Use this page to exclude properties of any branchable item type.

Item	Description
Item type:	Provides a list of the selected item types this VCM session whose properties you can exclude from a View Compare/Merge session. The item types in this list are determined by the choices of which item types you selected for View Compare/Merge in the VCM Wizard.
Available properties:	Lists all the properties available for the Item type selected. Select an item in the Available properties list and click the Exclude button to move it from Available properties to the Exclude properties list.
Exclude properties:	Lists all the properties selected for exclusion for the selected Item type. You can move an item from the Exclude properties list back to the Available properties list if you change your mind. Select the Item in the Exclude properties list and click the Include button to move it back to the Available properties list.

View Compare/Merge Session Filters

Filtering allows you to limit the types and numbers of folders that appear in the upper pane. The list of available filters depends on the perspective you are using in the VCM Session. StarTeam provides a set of predefined filters for View Compare/Merge which are listed below. The predefined filters are intended as starting points for you to create your own custom filters. Use the **Filter** drop-down list box on the toolbar to view and apply predefined file filters.

You can create additional filters for the **Merge Perspective** and the **Test Perspective**. To access the **Filters** dialog box, right-click a column header in the upper pane of the **Merge Perspective** or **Test Perspective** and choose **Filters**. The **Compare Perspective** has two predefined filter drop-down lists, but you cannot create any additional filters for that perspective.

For more details on using filters, see "Filtering Items in a View Compare/Merge Session" in the links below.

Below are the View Compare/Merge session predefined filters:

Item	Description
<Show Type>	Displays only items of the selected type in the Compare Perspective .
<Show Items With Differences>	Displays items from both the source and target views in the Compare Perspective that have differences.
<Show Unresolved Items>	Displays only items in the Compare Perspective with a merge status of Unresolved .
<Show All Items>	Displays all items of the selected type in the Compare Perspective .
Show Items: [action]	Displays only items in the Compare Perspective with the selected merge action.
<VCM: Items By Merge Status>	Displays items in the upper pane of the Merge Perspective sorted by merge status. This filter name is dynamic, depending on which component tab is selected. You can display items by merge status for files, change requests, and folders.
<VCM: Merge Properties>	Displays the items in the upper pane of the Merge Perspective with the primary descriptor in column one, and all the remaining VCM properties in the following columns.
Merge Action	Sorts items in the upper pane of the Merge Perspective and Test Perspective the by merge action.
<Show Items With Updates>	Displays only items from the source view in the Compare Perspective that will result in updates to the target.

Note: In the **Test Perspective**, you also see the filters for files when the **Files** tab is selected, for folders when the **Folders** tab is selected, and so on.

Related Concepts

[Filters](#)

Related Procedures

[Filtering Items in a View Compare/Merge Session](#)

[Creating Filters](#)

[Applying Predefined Filters](#)

[Copying Filters](#)

Change Package Properties

This topic presents the change package properties and their descriptions as displayed in the **Change Package Properties** dialog box. The **Change Package Properties** dialog box contains the following tabbed pages of properties.

Change Package

The following properties are available for **Change Packages**. User's with the proper access rights can change these properties.

Property	Description
Name	Displays the textual identification of the change package. The name must be unique among all other change packages for the same target view. By default, StarTeam generates a name that contains a timestamp, which helps prevent duplicate names. This property is the change package object's <i>primary descriptor</i> .
Description	Provides a full description of the change package. A typical use of the description property is to provide documentation not available in the change package's process item such as notes to testers, reviewers, or developers.
Session Type	Defines the change package's basic type. The type is set when the change package is first created and cannot be changed thereafter. It has the same values as VCM: Rebase, Replicate, and Promote.
Target View	Displays the name of the change package's target view.
Source View	Displays the name of the change package's source view.
Commit Time	Indicates the timestamp at which the change package was committed, or <code>null</code> if it has not yet been committed. This property cannot be directly modified by the user. Change packages that have been committed cannot be modified (except for the revision comment).
State	<p>Indicates the current workflow state of the change package. For the Hamachi release, this property will not be directly user-modifiable nor user-customizable. Furthermore, change packages will not be integrated with user-customizable workflow, hence the values of this property are defined by actions performed to the change package as it evolves. The initial enumeration values of the state property are:</p> <p>New: This is the state that every change package receives when it is first created, but before any updates are defined for it.</p> <p>Committed:: This state indicates that the change package has been committed, hence its changes have been applied. A committed change package and can no longer be modified (except for the revision comment).</p>
Transaction ID	Holds the ID of the transaction in which the change package's updates were applied. Prior to the <i>Committed</i> state, its value is <code>null</code> . Otherwise, the value can be used to query audit records in the Audit tab or commands used in the transaction by way of the StarTeam server command trace file.
Pre-commit View Label	Displays the name of the pre-modification revision label that was created in the target view. This property is <code>null</code> if this label type was not created.
Pre-commit Revision Label	Displays the name of the pre-modification revision label that was created in the target view. This property is <code>null</code> if this label type was not created.
Post-commit View Label	Displays the name of the post-modification view label that was created in the target view. This property is <code>null</code> if this label type was not created.
Post-commit Revision Label	Displays the name of the post-modification revision label that was created in the target view. This property is <code>null</code> if this label type was not created.

Committed by	Displays the name of the user who committed the change package or null if it has not yet been committed.
Committed In Build	<p>Displays the name of the first build label created in the same view after the change package is committed of null if it has not yet been committed.</p> <p>When a change package is committed, this property displays Next Build until a new build label is created.</p>
Responsibility	<p>Displays the name of the user who is currently responsible for the change package. It is initially set to the user that first creates the change package.</p> <p>If responsibility is changed to another user, and email notifications are enabled, the new user is notified of the change.</p>

Other Server Services

In addition to the StarTeam server's persistence and versioning service, change packages also use the server's locking service. This means that a change package can be locked exclusively or with a shared lock. An exclusive lock is automatically applied to a saved change session when it is opened for editing. A change session can also be opened in read-only mode, though a non-exclusive lock is not applied for this use.

Note: Change packages cannot be "flagged" (bookmarked).

Related Reference

[Change Package Properties](#)

Versioning Object Properties

Lists and defines the common properties of all objects that use versioning. For reference, these common properties are summarized below:

External (Internal) Property Name	Type	Description
Comment (Comment)	text	This calculated property provides the object's revision comment regardless of whether it is stored in the ShortComment or LCOMM table.
CommentID (CommentID)	int	If a revision comment < 2,000 characters exists, this field is a foreign key to the LCOMM table that contains the full comment.
Created By (CreatedUserID)	int	ID of the user that first created the object.
Created Time (CreatedTime)	timestamp	Date/time at which the object was created.
Deleted By (DeletedUserID)	int	ID of the user that deleted the object. -1 means the object is still "live".
Deleted Time (DeletedTime)	timestamp	Date/time that the object was deleted. 0 means object is still "live". Non-zero means the record is soft-deleted.
Dot Notation (DotNotation)	text	This calculated property is the dotted-notation representation of the object's version.
End Modified Time (EndModifiedTime)	timestamp	Date/time that this object revision was replaced by another revision. 0 identifies the latest (tip) revision.
Locked By (ExclusiveLocker)	int	ID of the user that currently has the object exclusively locked. -1 means the object is not exclusively locked.
Modified By (ModifiedUserID)	int	ID of the user that last modified the object.
Modified Time (ModifiedTime)	timestamp	Date/time that the object was last modified.
My Lock (MyLock)	int	This client-calculated property indicates if the current user has this object locked.
New Revision Comment (NewRevisionComment)	text	This client-calculated property serves as a place holder for a new revision comment value.
Non-Exclusive Lockers (NonExclusiveLockers)	text	List of users that currently have the object non-exclusively locked, if any.
Object ID (ID)	int	ID that is unique to each object in the database.
Read Only (ReadOnly)	int	This calculated property indicates if the object is currently read-only.
Revision Flags	int	<i>This field is currently always zero.</i>

(RevisionFlags)		
Short Comment	text	Value of the revision comment if it is less than 2,000 characters.
(ShortComment)		
Version	int	Revision number of this object revision. 0 = first revision.
(RevisionNumber)		

Related Reference

[Change Package Properties](#)

View Compare/Merge Terminology

This list defines common terms used in View Compare/Merge.

Term	Definition
Action	A strategy for resolving an item difference between a source view and target view. A default action may be chosen based on merge type and user-specified options, but the action may be changed by the user before it is enacted. There are many kinds of actions including ignore, share, reverse-share, delete, overwrite, merge, re-pin, and move. Combination actions are also possible for certain differences: move and merge, move and re-pin, and so on. For a description of each action, see the topic "View Compare/Merge Actions" in the Related Concepts links below.
Activity View	A branch-all variant (derived) child view whose purpose is to support a constrained-length development activity. For example, an enhancement or bug request is needed, so an activity view is created from the main view as of a specific milestone (for example, view label). Changes are made in the activity view, causing items to branch. Occasionally, changes made in the main view must be replicated to the activity view. Eventually, changes made in the activity view are "released" back to the main view, and the activity view becomes inactive.
Branch	<p>If a new revision of an item is created in a shared location while the Branch On Change property is on for the item, a new branch is created for that item and is reflected in the dot notation for the tip revision. Once the item has branched, it will not branch again unless it is shared again.</p> <p>Note: Only files, folders, and change requests can branch. Other item types do not branch.</p> <p>In the location where an item was first created, it cannot branch. When an item is shared, it can acquire branching behavior. By default, the shared item acquires the branching behavior set by the Branch On Change property for the view, but this property can be reset manually for each item. If the behavior is set to Branch On Change, the next revision will branch, adding a new pair of integers to its dot notation.</p> <p>If you compare the dot notation for two revisions, you can determine if they are on the same branch or on different branches. If the two dot notations have the same number of integers and those integers are identical (except for the last integer), then the revisions are on the same branch. A difference in the number of integers, or a difference in the value of any integer except the last, indicates that two matching items are on different branches. For example, 1.7 and 1.7.1.0 are not on the same branch. 1.7.1.0 and 1.7.2.0 are not on the same branch. 1.7 and 1.9 are on the same branch. 1.7.1.8 and 1.7.1.15 are on the same branch.</p>
Common Ancestor	The common ancestor of two matching items is the most recent revision of the item to appear in the histories of both items, or if merge points are being used, the source revision for which the merge point has been set. For example, suppose that the root view is the source view, one of the branching child views of the root is the target view, and merge points have not been set. If the item in the source has the dot notation 1.9 and the item in the target is 1.7.1.2, the common ancestor is 1.7 and probably was the tip revision at the time that the child view was created. View Compare/Merge uses the common ancestor of the matching items to determine if changes have been made to the source and target items.
Conflict	The properties or contents of two matching items require manual intervention in a merge situation. For file contents, conflicts arise if two matching blocks or lines of code have both been changed since the common ancestor or the most recent merge point. For properties, if a property has changed in both matching items, that property produces a conflict that must be resolved manually. If the property value of only one of the items has changed, that property does not have a conflict. If only the source property has changed, the difference for that property can be automatically resolved by overwriting the property value in the target with the property value in the source. If only the target property has changed, the difference is ignored.

Difference	Used to describe an item that is not identical in both the source view and the target view. For example, the item could be present in one view and not the other; it could be in both views but in different folders; it could be in both views but have different revisions, and so on.
Dot Notation	<p>A set of pairs of integers separated by periods indicating the branch that each revision is on and its place on that branch.</p> <p>For example, suppose you add a file to a root view. The first revision of that file has the dot notation 1.0; the second revision has the dot notation 1.1, and so on. Suppose you create two branching views as children of the root view, and both child views include the file <code>info.htm</code>, and the tip revision of <code>info.htm</code> is 1.7 at the time the child views were created. Both child views will show revisions 1.0 through 1.7 in the history for <code>info.htm</code>. The history also indicates that those revisions were created in the parent view. Now suppose that someone changes <code>info.htm</code> in the first child view and checks in the file. The new revision (the first revision of <code>info.htm</code> in that particular child view) will have the dot notation 1.7.1.0. This is a new branch. To continue with the same example as before, suppose that a developer makes a different set of changes to <code>info.htm</code> in the second child view. As those changes are checked in, the dot notation of the new revision (the first revision of <code>info.htm</code> in the second child view for this file) becomes 1.7.2.0 and starts a new branch. The dot notation cannot become 1.7.1.0 because that branch has already been created in the first child view.</p>
Floating	Shared items not configured to a specific label, promotion state, or timestamp are said to be floating. For example, suppose an item with the revision 1.7 was shared into a second view and allowed to float. As the original item moves to revisions 1.8 and 1.9 , so does the shared item. When pinned, a share stops floating and stays at the revision number it is pinned to, unless it branches.
Ignore	View Compare/Merge takes no action. It is always available as the action to be performed on any pair of matching items. It is sometimes the default action and sometime the only action available. For example, if an item has been deleted from the source, but its matching item is not only in use but has changed in the target since the last merge, View Compare/Merge expects that you will want to keep this item in the target view. The default action in this case is to ignore the differences between the two matching items.
Merge	When this action is selected, View Compare/Merge merges the properties of the matching items and creates a new revision in the target view. The merge can be done automatically or manually. If there are conflicts, the merge must be manual. View Compare/Merge also sets a merge point for future reference.
Merge Point	A merge point indicates that a merge occurred between a particular revision of an item in the source view and a particular revision of its matching item in the target view. Merge points are directional in that they denote which item was the source and which was the target. It takes precedence over the use of the common ancestor. In future comparisons and merges, if neither the source nor target item has changed since the merge point was set, View Compare/Merge takes no action. A merge point is set for any Merge , Overwrite , or Mark As Resolved action.
Merge Type	The overall view compare/merge strategy. The possible merge types are Rebase, Promote, and Replicate.
Promote	A Promote updates a parent view to reflect the changes made in one of its children. Changes in the child view usually result in branching, so changes are usually merged into the parent view. If the item exists only in the child, it can be reverse-shared into the parent. A reverse-share is a good idea because the parent view is the best location for an item that is the root of a reference tree.
Rebase	<p>A Rebase operation updates a child view to reflect changes made in its parent view. Items that have not changed in the child are rebased simply by re-pinning the item in the child view to match the current revision of the matching item in the parent.</p> <p>Example of a Rebase: Suppose you create a child view based on the root view one week and wish you had waited. Several changes have gone into the root view in the past week,</p>

but little or no work has occurred in the child. A **Rebase** operation changes the child view so that each item is re-pinned to the current revision in the parent. If an item is new in the parent, it is shared into the child. If an item has branched in the child, re-pinning is not possible and a merge is necessary.

Release View	A branch-all variant (derived) child view whose purpose is to support continuing maintenance activities for a specific product release. A release view is created from the main view as of a specific milestone such as a view label. Changes to the release view support bug fixes or minor enhancements. Some of these changes must be replicated back to the main view or other release views. Some changes made to the main view must be replicated to one or more release views.
Re-pin	An item in a target view can be re-pinned if the two matching items are on the same branch. The act of re-pinning changes the common ancestor for a pair of matching items. It also moves the tip revision in the target view forward at least one revision. For example, if the item in the source view is 1.9 and the item in the target view is 1.7, re-pinning changes the tip revision in the target view to the 1.9 revision. It is equivalent to setting the configuration time of the item forward to a time after the creation of the 1.9 revision in the source view. The common ancestor for the two items is now revision 1.9. Re-pinning is most commonly done in a Rebase operation when the source view is the parent of the child view.
Replicate	For views that are not parent/child, View Compare/Merge expects you to perform a Replicate operation. The actions available during a Replicate operation and the default action select by View Compare/Merge are the same as for a Rebase operation. However, while re-pinning may be the most used action in a Rebase operation, it may be the least used action in a Replicate operation. This is because, depending on the relationship between the views, fewer of the items will be on the same branch.
Reverse-Share	<p>When an item is reverse-shared, it is moved from the source view to the target view and then shared back into the source. The items in both views have the same data, the same number of revisions, the same dot notation, and so on. However, the reference tree changes: the item in the source view becomes the child of the item in the reference tree of the target view, reversing the previous parent/child relationship. The only differences between the original item in the source view and that item as newly shared back into the source view is its placement in the reference tree and its Branch On Change status. That status is on or off, depending on the default setting for Branch On Change in the source view. The item in the target is identical to the item that was originally in the source, although its location has changed. It has the same properties and the same place in the reference tree.</p> <p>For example, when an item M is shared from folder A to B, M is simply copied from A to B, creating M' in B as a child share of M. In a reverse-share from A to B, M is moved from A to B and then copied (shared) back to A, creating M' in A as a child share of M. This technique "transfers ownership" of the item from A to B.</p>
Share	An item is shared manually by performing a share (drag-and-drop) operation. An item is shared automatically when a child view is created. Shared items can branch. When an item is shared, either manually or automatically, from one location to another, the item in the new location has the same data, the same number of revisions, the same dot notation, and so on. However, a new reference to this item appears in the reference tree for this item. The only differences between the original item and the shared item are their locations and, perhaps, their Branch On Change status. That status is on or off, depending on the default setting for Branch On Change in the receiving view. Later, the Branch On Change status can be changed manually. When a shared item with its Branch On Change status is changed, it branches and the Branch On Change status is disabled.
Source View	A view that is compared with a target view, and whose changes are potentially merged to the target view. In some cases, the source view can use a rolled-back and/or read-only configuration.
Target Types	The item types that are to be considered during a view compare/merge operation.

Target View	A view that is compared with a source view and which potentially receives changes from the source view. The target view must be able to be updated, and it must be a variant (branching) view or a non-derived (blank) view.
-------------	--

Related Concepts

[Overview of View Compare/Merge \(VCM\)](#)
[View Compare/Merge Actions](#)
[View Compare/Merge Session Perspectives](#)
[VCM Merge Types, Rules, and Scenarios](#)
[View Merge Type Scenarios](#)

Related Procedures

[Comparing and Merging Views](#)
[Comparing and Merging Views](#)
[Changing the View Compare/Merge Session Working Folder](#)
[Running View Compare/Merge from the Command-line](#)

Related Reference

[View Compare/Merge Wizard](#)

Cross-Platform Client Menus

This section contains reference topics related to the menu commands found in the StarTeam Cross-Platform Client.

In This Section

[Project Menu](#)

This topic describes the menu commands found on the **Project** menu of the Cross-Platform Client.

[View Menu](#)

This topic describes the menu commands found on the **View** menu of the Cross-Platform Client.

[Folder Tree Menu](#)

This topic describes the menu commands found on the **Folder Tree** menu of the Cross-Platform Client.

[Folder Menu](#)

This topic describes the menu commands found on the **Folder** menu of the Cross-Platform Client.

[File Menu](#)

This topic describes the menu commands found on the **File** menu of the Cross-Platform Client.

[Change Request Menu](#)

This topic describes the menu commands found on the **Change Request** menu of the Cross-Platform Client.

[Requirement Menu](#)

This topic describes the menu commands found on the **Requirement** menu of the Cross-Platform Client.

[Task Menu](#)

This topic describes the menu commands found on the **Task** menu of the Cross-Platform Client.

[Topic Menu](#)

This topic describes the menu commands found on the **Topic** menu of the Cross-Platform Client.

[Audit Menu](#)

This topic describes the menu commands found on the **Audit** menu of the Cross-Platform Client.

[Tools Menu](#)

This topic describes the menu commands found on the **Tools** menu of the Cross-Platform Client.

[Window Menu](#)

This topic describes the menu commands found on the **Window** menu of the Cross-Platform Client.

[Help Menu](#)

This topic describes the menu commands found on the **Help** menu of the Cross-Platform Client.

Project Menu

This topic describes the menu commands found on the [Project](#) menu of the Cross-Platform Client.

The [Project](#) menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

Project Menu Command	Description
New	Starts the New Project Wizard , and creates a new project.
Open	Starts the Open Project Wizard , and opens an existing project.
Close	Closes the current project.
Open Shortcut	Opens the project view associated with the shortcut in the configuration it had when the shortcut was created. The shortcut displays each view in the configuration it had at the time the shortcut was created.
Save Shortcut As	Creates a new shortcut to the current project view. Shortcuts are usually saved to your desktop.
Open StarTeam URL	Opens StarTeam URL shortcut links to projects, views, folders, not-in-view folders, and items. This enables you to quickly access specific locations in a project.
Recent Projects	Reopens a recently-opened project view.
Properties	Opens the Project Properties dialog box for the current project.
Access Rights	Opens the Project Access Rights dialog box for the current project.
Delete	Deletes a project.
Connection Properties	Opens the Connection Properties dialog box for the current project and displays the current server description, address, endpoint, user name, and logon time for the connection.
Log Off	Closes all the views on the server that you have logged into during this session.
Exit	Exits the application.

Related Procedures

[Managing Projects](#)

Related Reference

[Cross-Platform Client Menus](#)

View Menu

This topic describes the menu commands found on the [View](#) menu of the Cross-Platform Client.

The [View](#) menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

View Menu Command	Description
New	Starts the New View Wizard , and creates a new view.
Delete	Deletes a view from the current project.
Select View	Opens a dialog box displaying the view tree for the current project where you can choose to launch another view.
Select Configuration	Opens the Select a View Configuration dialog box where you can specify which files to check out in the current view based on a label, promotion state, or a date and time.
Properties	Opens the View Properties dialog box for the current view.
Access Rights	Opens the View Access Rights dialog box for the current view.
Labels	Opens the Labels dialog box where you can create, delete, or freeze (lock) a view label or a revision label or review its properties.
Promotion	Opens the Promotion dialog box where you can create, edit, delete, reorder, or set access rights for promotion states. A promotion state is a way of grouping files as they progress through different stages to completion (for example, Develop, Test, and Production).
Compare/Merge	Opens the View Compare/Merge Wizard where you can compare or merge different views.

Related Procedures

[Managing Views](#)

Related Reference

[Cross-Platform Client Menus](#)

Folder Tree Menu

This topic describes the menu commands found on the **Folder Tree** menu of the Cross-Platform Client.

The **Folder Tree** menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

Folder Tree Menu Command	Description
New	Starts the New Folder Wizard , and creates a new folder in the current view.
Add to View	Adds the selected local folder to the current view.
Delete	Deletes the folder and any subfolders from the StarTeam repository. You can optionally delete the local working folders.
Create Working Folders	Creates the path to a working folder on your hard drive. Otherwise, if the path to a working folder does not exist on your hard drive, you must create it manually before adding files or checking in files from that working folder.
Delete Local Folders	Deletes the specified local folder and any subfolders. This action does not delete any folder or subfolder from the StarTeam repository.
Open Local Folder	Opens a local folder in a file browser.
Show Not-In-View Folders	Works as a toggle. Choose to make not-in-view folders visible to the client. Not-in-view folders reside in your local working folder, but they have not been added to the StarTeam repository.
Lock/Unlock	Opens the Set My Lock Status dialog box where you can specify the lock status (unlocked, exclusive, or non-exclusive) for the folder or break an existing lock on a folder.
Properties	Opens the Folder Properties dialog box for the selected folder.
Labels	Opens the Labels dialog box where you can assign labels to and/or removes labels from folders.
Advanced ▶ Behavior	Opens the Folder Behavior dialog box where you can view or change the behavior of the selected folder.
Advanced ▶ References	Opens the Folder References dialog box where you can view the relationships between an original folder or item and the others based on it.
Advanced ▶ Access Rights	Opens the Folder Access Rights dialog box for the selected folder.
Advanced ▶ View Compare/Merge	Opens the View Compare/Merge Wizard .
Links ▶ Create Link	Start a link from a selected folder to another item.
Links ▶ Complete Link	Complete the link from a folder to the selected item.
Links ▶ Cancel Link	Stop the process of creating a link.
Copy URL to Clipboard	Places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.

Related Procedures

[Working with Folders and Items](#)

Related Reference

[Cross-Platform Client Menus](#)

Folder Menu

This topic describes the menu commands found on the **Folder** menu of the Cross-Platform Client.

The **Folder** menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

Folder Menu Command	Description
New	Starts the New Folder Wizard , and creates a new folder in the current view.
Add to View	Adds the selected local folder to the current view.
Create Working Folders	Creates the path to a working folder on your hard drive. Otherwise, if the path to a working folder does not exist on your hard drive, you must create it manually before adding files or checking in files from that working folder.
Delete Local Folders	Deletes the specified local folder and any subfolders. This action does not delete any folder or subfolder from the StarTeam repository.
Open Local Folder	Opens a local folder in a file browser.
Properties	Opens the Folder Properties dialog box for the selected folder.
Copy URL to Clipboard	Places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.
Send To	E-mail a copy of the folder properties to one or more team members.
Save Shortcut	Creates a shortcut to the selected folder. You can use the shortcut to start the application, open the project view in the configuration in which the shortcut was created, and open the Properties dialog box for the selected folder.
Lock/Unlock	Opens the Set My Lock Status dialog box where you can specify the lock status (unlocked, exclusive, or non-exclusive) for the folder or break an existing lock on a folder.
Compare Properties	Compares the properties of two selected folders.
Labels ▶ New	Opens the Attach a New Revision Label dialog box where you can create a new revision label and attach it to the selected folder or folders.
Labels ▶ Attach	Opens the Attach a Label dialog box where you can add an existing label to the selected folder or folders.
Labels ▶ Detach	Opens the Detach a Label dialog box where you can remove a label from the selected folder or folders.
Advanced ▶ Export	Opens the Export dialog box where you can export the data displayed in the upper pane for use in a spreadsheet, database, or another application.
Advanced ▶ Behavior	Opens the Folder Behavior dialog box where you can view or change the behavior of the selected folder.
Advanced ▶ Item Access Rights	Opens the Folder Access Rights dialog box for the selected folder.
Advanced ▶ Component Access Rights	Opens the Folder Component Access Rights dialog box.
Advanced ▶ View Compare/Merge	Opens the View Compare/Merge Wizard .
Select ▶ All	Selects all folders displayed in the upper pane.
Select ▶ By Query	Opens the Select Query dialog box where you can specify a query that selects all items displayed in the upper pane that match the specified query.
Select ▶ By Label	Opens the Select a Label dialog box where you can specify a label that selects all items displayed in the upper pane that have the specified label.

All Descendants	Displays all child folders associated with the folder selected from the folder tree. When this option is not selected, the upper pane displays only those folders associated with the folder selected in the folder tree.
Delete	Deletes the folder and any subfolders from the StarTeam repository. You can optionally delete the local working folders.
Find	Finds all folders containing text that matches the specified text string.
Find Next	Finds the next folder containing text that matches the specified text string.
Find Previous	Finds the previous folder containing text that matches the specified text string.
Filters ► Show Fields	Opens the Show Fields dialog box where you can select which column headers to display in the upper pane.
Filters ► Sort and Group	Opens the Sort and Group dialog box where you can specify how to sort and group the data in the upper pane based on up to four folder properties.
Filters ► Queries	Opens the Queries dialog box where you can create or apply a query. Items that match the query become the rows displayed in the upper pane.
Filters ► Save Current Settings	Opens the Save Current Settings dialog box where you can save the current column headers, query, and method of sorting and grouping as a filter, which can be reapplied in the future.
Filters ► Reset Current Settings	Returns to the current default filter as defined on the server.
Filters ► Filters	Opens the Filters dialog box where you can display the filters available for this component. This option allows you to select a filter or create a new one. An asterisk preceding a filter name in the filter list box indicates that you have changed the contents of the upper pane since the filter was applied. For example, you may have displayed additional fields, grouped the items differently, or applied another query.
Reports	Opens the Reports dialog box where you can create a report using the selected folder.
Charts ► Simple	Creates a simple chart. A simple chart contains only one series.
Charts ► Correlation	Creates a correlation chart. A correlation chart shows the relationship or degree of relationship between numeric values in several series.
Charts ► Time-series	Creates a time-series chart. A time-series chart is a line chart that shows the number of items that have the same day, week, or month in the specified time/ date field.
Links ► Create Link	Start a link from a selected folder to another item.
Links ► Complete Link	Complete the link from a folder to the selected item.
Links ► Cancel Link	Stop the process of creating a link.
Linked Files ► Check In All	Checks in any files from the current view linked to the item.
Linked Files ► Check Out All	Checks out any files from the current view linked to the item.
Linked Files ► Select All	Selects all the files associated with a linked item in the current view.
Linked Files ► Add to Selection	Opens the upper pane to the File pane, activates the All Descendants View, and adds all the files linked to the item to the existing file selection.

Related Procedures

[Working with Folders and Items](#)

Related Reference

[Cross-Platform Client Menus](#)

File Menu

This topic describes the menu commands found on the **File** menu of the Cross-Platform Client.

The **File** menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

File Menu Command	Description
Open	Opens the selected file in its associated application or starts an executable file. If you do not have a copy of the file locally, the application prompts you to save a copy in your local working directory.
Edit	Edits the selected file in the default text editor used by your system or by the editor specified in the Alternate Applications dialog box.
Open Containing Folder	Opens the local folder that contains the file in a file browser.
Properties	Opens the File Properties dialog box for the selected file.
Copy URL to Clipboard	Places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.
Send To	E-mail a copy of the file properties to one or more team members.
Save Shortcut	Creates a shortcut to the selected file. You can use the shortcut to start the application, open the project view in the configuration in which the shortcut was created, and open the Properties dialog box for the selected file.
Add Files	Adds new files to the current project view.
Check In	Checks the selected files into the current project view.
Check Out	Copies the selected files to the appropriate working folders.
Check Out All	Checks out all the files in the branch of the application folder hierarchy that has the selected folder as its root folder.
Update Status	Updates the status of the selected files. If no files are selected, all file statuses are updated.
Lock/Unlock	Opens the Set My Lock Status dialog box where you can specify the lock status (unlocked, exclusive, or non-exclusive) for the folder or break an existing lock on a file.
Compare Properties	Compares the properties of two selected files.
Compare Contents	Opens File Compare/Merge.
Labels ▶ New	Opens the Attach a New Revision Label dialog box where you can create a new revision label and attach it to the selected file or files.
Labels ▶ Attach	Opens the Attach a Label dialog box where you can add an existing label to the selected file or files.
Labels ▶ Detach	Opens the Detach a Label dialog box where you can remove a label from the selected file or files.
Advanced ▶ Convert Archive	Converts files saved in the Native-I vault structure to the Native-II vault structure. Native-I vaults were last available in StarTeam 2005 Release 2.
Advanced ▶ Export	Opens the Export dialog box where you can export the data displayed in the upper pane for use in a spreadsheet, database, or another application.
Advanced ▶ Behavior	Opens the File Behavior dialog box where you can view or change the behavior of the selected file.

Advanced ▶ Item Access Rights	Opens the File Access Rights dialog box for the selected file.
Advanced ▶ Component Access Rights	Opens the File Component Access Rights dialog box.
Advanced ▶ Customize	Opens the Customize dialog box where you can create custom fields that become file properties.
Advanced ▶ View Compare/Merge	Opens the View Compare/Merge Wizard .
Select ▶ All	Selects all files displayed in the upper pane.
Select ▶ By Query	Opens the Select Query dialog box where you can specify a query that selects all items displayed in the upper pane that match the specified query.
Select ▶ By Label	Opens the Select a Label dialog box where you can specify a label that selects all items displayed in the upper pane that have the specified label.
Expand All	Expands all the groups in the files list.
Collapse All	Collapses all the groups in the files list.
All Descendants	Displays all files associated with the folder selected from the folder hierarchy and all of its child folders. When this option is not selected, the upper pane displays only those files associated with the selected folder.
Flag	Flags an item. Only you can see and remove your flags. When an item is flagged, Yes displays in the Flag column in the upper pane. Note that this column does not display by default. Choose File ▶ Filters ▶ Show Fields to add the column to the upper pane.
Remove Flag	Removes the flag from an item.
Delete	Deletes the file from the StarTeam repository. You can optionally delete the local file.
Find	Finds all files containing text that matches the specified text string.
Find Next	Finds the next files containing text that matches the specified text string.
Find Previous	Finds the previous file containing text that matches the specified text string.
Filters ▶ Show Fields	Opens the Show Fields dialog box where you can select which column headers to display in the upper pane.
Filters ▶ Sort and Group	Opens the Sort and Group dialog box where you can specify how to sort and group the data in the upper pane based on up to four file properties.
Filters ▶ Queries	Opens the Queries dialog box where you can create or apply a query. Items that match the query become the rows displayed in the upper pane.
Filters ▶ Save Current Settings	Opens the Save Current Settings dialog box where you can save the current column headers, query, and method of sorting and grouping as a filter, which can be reapplied in the future.
Filters ▶ Reset Current Settings	Returns to the current default filter as defined on the server.
Filters ▶ Filters	Opens the Filters dialog box where you can display the filters available for this component. This option allows you to select a filter or create a new one. An asterisk preceding a filter name in the filter list box indicates that you have changed the contents of the upper pane since the filter was applied. For example, you may have displayed additional fields, grouped the items differently, or applied another query.
Reports	Opens the Reports dialog box where you can create a report using the selected file.
Charts ▶ Simple	Creates a simple chart. A simple chart contains only one series.
Charts ▶ Distribution	Creates a distribution chart. A distribution chart is the same as a pie chart. Each wedge indicates what fraction of the whole a group represents.

Charts ► Correlation	Creates a correlation chart. A correlation chart shows the relationship or degree of relationship between numeric values in several series.
Charts ► Time-series	Creates a time-series chart. A time-series chart is a line chart that shows the number of items that have the same day, week, or month in the specified time/date field.
Links ► Create Link	Start a link from a selected file to another item.
Links ► Complete Link	Complete the link from a file to the selected item.
Links ► Cancel Link	Stop the process of creating a link.
Linked Files ► Check In All	Checks in any files from the current view linked to the item.
Linked Files ► Check Out All	Checks out any files from the current view linked to the item.
Linked Files ► Select All	Selects all the files associated with a linked item in the current view.
Linked Files ► Add to Selection	Opens the upper pane to the File pane, activates the All Descendants View, and adds all the files linked to the item to the existing file selection.

Related Procedures

[Managing Files](#)

Related Reference

[Cross-Platform Client Menus](#)

Change Request Menu

This topic describes the menu commands found on the **Change Request** menu of the Cross-Platform Client.

The **Change Request** menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

Change Request Menu Command	Description
New	Creates a new change request.
Properties	Opens the Change Request Properties dialog box for the selected change request.
Copy URL to Clipboard	Places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.
Send To	E-mail a copy of the change request properties to one or more team members.
Save Shortcut	Creates a shortcut to the selected change request. You can use the shortcut to start the application, open the project view in the configuration in which the shortcut was created, and open the Properties dialog box for the selected change request.
Lock/Unlock	Opens the Set My Lock Status dialog box where you can specify the lock status (unlocked, exclusive, or non-exclusive) for the folder or break an existing lock on a change request.
Compare Properties	Compares the properties of two selected change requests.
Labels ▶ New	Opens the Attach a New Revision Label dialog box where you can create a new revision label and attach it to the selected change request or change requests.
Labels ▶ Attach	Opens the Attach a Label dialog box where you can add an existing label to the selected change request or change requests.
Labels ▶ Detach	Opens the Detach a Label dialog box where you can remove a label from the selected change request or change requests .
Advanced ▶ Export	Opens the Export dialog box where you can export the data displayed in the upper pane for use in a spreadsheet, database, or another application.
Advanced ▶ Behavior	Opens the Change Request Behavior dialog box where you can view or change the behavior of the selected change request .
Advanced ▶ Item Access Rights	Opens the Change Request Access Rights dialog box for the selected change request .
Advanced ▶ Component Access Rights	Opens the Change Request Component Access Rights dialog box.
Advanced ▶ Customize	Opens the Customize dialog box where you can create custom fields that become change request properties.
Advanced ▶ View Compare/Merge	Opens the View Compare/Merge Wizard .
Select ▶ All	Selects all change requests displayed in the upper pane.
Select ▶ By Query	Opens the Select Query dialog box where you can specify a query that selects all items displayed in the upper pane that match the specified query.
Select ▶ By Label	Opens the Select a Label dialog box where you can specify a label that selects all items displayed in the upper pane that have the specified label.
Expand All	Expands all the groups in the change request list.
Collapse All	Collapses all the groups in the change request list.

All Descendants	Displays all change requests associated with the folder selected from the folder hierarchy and all of its child folders. When this option is not selected, the upper pane displays only those change requests associated with the selected folder.
Mark as Read	Displays the selected change requests in regular font (not bold), which indicates that they have been read.
Mark as Unread	Displays the selected change requests in bold font, which indicates that they have not been read.
Flag	Flags an item. Only you can see and remove your flags. When an item is flagged, Yes displays in the Flag column in the upper pane. Note that this column does not display by default. Choose Change Request ▸ Filters ▸ Show Fields to add the column to the upper pane.
Remove Flag	Removes the flag from an item.
Delete	Deletes the change request from the StarTeam repository.
Find	Finds all change requests containing text that matches the specified text string.
Find Next	Finds the next change request containing text that matches the specified text string.
Find Previous	Finds the previous change request containing text that matches the specified text string.
Filters ▸ Show Fields	Opens the Show Fields dialog box where you can select which column headers to display in the upper pane.
Filters ▸ Sort and Group	Opens the Sort and Group dialog box where you can specify how to sort and group the data in the upper pane based on up to four change request properties.
Filters ▸ Queries	Opens the Queries dialog box where you can create or apply a query. Items that match the query become the rows displayed in the upper pane.
Filters ▸ Save Current Settings	Opens the Save Current Settings dialog box where you can save the current column headers, query, and method of sorting and grouping as a filter, which can be reapplied in the future.
Filters ▸ Reset Current Settings	Returns to the current default filter as defined on the server.
Filters ▸ Filters	Opens the Filters dialog box where you can display the filters available for this component. This option allows you to select a filter or create a new one. An asterisk preceding a filter name in the filter list box indicates that you have changed the contents of the upper pane since the filter was applied. For example, you may have displayed additional fields, grouped the items differently, or applied another query.
Reports	Opens the Reports dialog box where you can create a report using the selected change request.
Charts ▸ Simple	Creates a simple chart. A simple chart contains only one series.
Charts ▸ Distribution	Creates a distribution chart. A distribution chart is the same as a pie chart. Each wedge indicates what fraction of the whole a group represents.
Charts ▸ Correlation	Creates a correlation chart. A correlation chart shows the relationship or degree of relationship between numeric values in several series.
Charts ▸ Time-series	Creates a time-series chart. A time-series chart is a line chart that shows the number of items that have the same day, week, or month in the specified time/date field.
Links ▸ Create Link	Start a link from a selected change request to another item.
Links ▸ Complete Link	Complete the link from a change request to the selected item.
Links ▸ Cancel Link	Stop the process of creating a link.

Set Active Process Item ▶ Current View	Designates the selected change request in the current view as the active process item. Active process items are used when you add change requests to a view or check them in.
Set Active Process Item ▶ Select View	Designates the selected change request in an alternate view as the active process item. Active process items are used when you add change requests to a view or check them in.
Clear Active Process Item	Changes the selected change request from active to inactive. Active process items are used when you add change requests to a view or check them in.

Related Procedures

[Working with Change Requests](#)

Related Reference

[Cross-Platform Client Menus](#)

Requirement Menu

This topic describes the menu commands found on the **Requirement** menu of the Cross-Platform Client.

The **Requirement** menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

Requirement Menu Command	Description
New	Creates a new requirement.
New Child Requirement	Creates a requirement that is the child of another requirement.
Properties	Opens the Requirement Properties dialog box for the selected requirement.
Copy URL to Clipboard	Places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.
Send To	E-mail a copy of the requirement properties to one or more team members.
Save Shortcut	Creates a shortcut to the selected requirement. You can use the shortcut to start the application, open the project view in the configuration in which the shortcut was created, and open the Properties dialog box for the selected requirement.
Lock/Unlock	Opens the Set My Lock Status dialog box where you can specify the lock status (unlocked, exclusive, or non-exclusive) for the folder or break an existing lock on a requirement.
Compare Properties	Compares the properties of two selected requirements.
Labels ▶ New	Opens the Attach a New Revision Label dialog box where you can create a new revision label and attach it to the selected requirement or requirements.
Labels ▶ Attach	Opens the Attach a Label dialog box where you can add an existing label to the selected requirement or requirements.
Labels ▶ Detach	Opens the Detach a Label dialog box where you can remove a label from the selected requirement or requirements.
Advanced ▶ Export	Opens the Export dialog box where you can export the data displayed in the upper pane for use in a spreadsheet, database, or another application.
Advanced ▶ Behavior	Opens the Requirement Behavior dialog box where you can view or change the behavior of the selected requirement.
Advanced ▶ Item Access Rights	Opens the Requirement Access Rights dialog box for the selected requirement.
Advanced ▶ Component Access Rights	Opens the Requirement Component Access Rights dialog box.
Advanced ▶ Customize	Opens the Customize dialog box where you can create custom fields that become requirement properties.
Advanced ▶ View Compare/Merge	Opens the View Compare/Merge Wizard .
Select ▶ All	Selects all requirements displayed in the upper pane.
Select ▶ By Query	Opens the Select Query dialog box where you can specify a query that selects all items displayed in the upper pane that match the specified query.
Select ▶ By Label	Opens the Select a Label dialog box where you can specify a label that selects all items displayed in the upper pane that have the specified label.
Expand All	Expands all the groups in the requirements list.
Collapse All	Collapses all the groups in the requirements list.

All Descendants	Displays all requirements associated with the folder selected from the folder hierarchy and all of its child folders. When this option is not selected, the upper pane displays only those requirements associated with the selected folder.
Mark as Read	Displays the selected requirements in regular font (not bold), which indicates that they have been read.
Mark as Unread	Displays the selected requirements in bold font, which indicates that they have not been read.
Mark Thread as Read	Displays the entire thread in regular type, which indicates that the requirements it contains have been read.
Mark Thread as Unread	Displays the entire thread in boldface type, which indicates that the requirements it contains have not been read.
Flag	Flags an item. Only you can see and remove your flags. When an item is flagged, Yes displays in the Flag column in the upper pane. Note that this column does not display by default. Choose Requirement ▸ Filters ▸ Show Fields to add the column to the upper pane.
Remove Flag	Removes the flag from an item.
Delete	Deletes the requirement from the StarTeam repository.
Find	Finds all requirements containing text that matches the specified text string.
Find Next	Finds the next requirement containing text that matches the specified text string.
Find Previous	Finds the previous requirement containing text that matches the specified text string.
Filters ▸ Show Fields	Opens the Show Fields dialog box where you can select which column headers to display in the upper pane.
Filters ▸ Sort and Group	Opens the Sort and Group dialog box where you can specify how to sort and group the data in the upper pane based on up to four requirement properties.
Filters ▸ Queries	Opens the Queries dialog box where you can create or apply a query. Items that match the query become the rows displayed in the upper pane.
Filters ▸ Save Current Settings	Opens the Save Current Settings dialog box where you can save the current column headers, query, and method of sorting and grouping as a filter, which can be reapplied in the future.
Filters ▸ Reset Current Settings	Returns to the current default filter as defined on the server.
Filters ▸ Filters	Opens the Filters dialog box where you can display the filters available for this component. This option allows you to select a filter or create a new one. An asterisk preceding a filter name in the filter list box indicates that you have changed the contents of the upper pane since the filter was applied. For example, you may have displayed additional fields, grouped the items differently, or applied another query.
Reports	Opens the Reports dialog box where you can create a report using the selected requirement.
Charts ▸ Simple	Creates a simple chart. A simple chart contains only one series.
Charts ▸ Distribution	Creates a distribution chart. A distribution chart is the same as a pie chart. Each wedge indicates what fraction of the whole a group represents.
Charts ▸ Correlation	Creates a correlation chart. A correlation chart shows the relationship or degree of relationship between numeric values in several series.
Charts ▸ Time-series	Creates a time-series chart. A time-series chart is a line chart that shows the number of items that have the same day, week, or month in the specified time/date field.

Links ▶ Create Link	Start a link from a selected requirement to another item.
Links ▶ Complete Link	Complete the link from a requirement to the selected item.
Links ▶ Cancel Link	Stop the process of creating a link.
Set Active Process Item ▶ Current View	Designates the selected requirement in the current view as the active process item. Active process items are used when you add requirements to a view or check them in.
Set Active Process Item ▶ Select View	Designates the selected requirement in an alternate view as the active process item. Active process items are used when you add requirements to a view or check them in.
Clear Active Process Item	Changes the selected requirement from active to inactive. Active process items are used when you add requirements to a view or check them in.

Related Procedures

[Using Requirements](#)

Related Reference

[Cross-Platform Client Menus](#)

Task Menu

This topic describes the menu commands found on the **Task** menu of the Cross-Platform Client.

The **Task** menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

Task Menu Command	Description
New	Creates a new task.
New Subtask	Creates a task that is the child of another task.
Add Work	Adds a work record to the selected task. You can add work only to tasks that have no subtasks.
Properties	Opens the Task Properties dialog box for the selected task.
Copy URL to Clipboard	Places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.
Send To	E-mail a copy of the task properties to one or more team members.
Save Shortcut	Creates a shortcut to the selected task. You can use the shortcut to start the application, open the project view in the configuration in which the shortcut was created, and open the Properties dialog box for the selected task.
Lock/Unlock	Opens the Set My Lock Status dialog box where you can specify the lock status (unlocked, exclusive, or non-exclusive) for the folder or break an existing lock on a task.
Compare Properties	Compares the properties of two selected tasks.
Labels ▶ New	Opens the Attach a New Revision Label dialog box where you can create a new revision label and attach it to the selected task or tasks.
Labels ▶ Attach	Opens the Attach a Label dialog box where you can add an existing label to the selected task or tasks.
Labels ▶ Detach	Opens the Detach a Label dialog box where you can remove a label from the selected task or tasks.
Advanced ▶ Export	Opens the Export dialog box where you can export the data displayed in the upper pane for use in a spreadsheet, database, or another application.
Advanced ▶ Behavior	Opens the Task Behavior dialog box where you can view or change the behavior of the selected task.
Advanced ▶ Item Access Rights	Opens the Task Access Rights dialog box for the selected task.
Advanced ▶ Component Access Rights	Opens the Task Component Access Rights dialog box.
Advanced ▶ Customize	Opens the Customize dialog box where you can create custom fields that become task properties.
Advanced ▶ View Compare/Merge	Opens the View Compare/Merge Wizard .
Select ▶ All	Selects all tasks displayed in the upper pane.
Select ▶ By Query	Opens the Select Query dialog box where you can specify a query that selects all items displayed in the upper pane that match the specified query.
Select ▶ By Label	Opens the Select a Label dialog box where you can specify a label that selects all items displayed in the upper pane that have the specified label.
Expand All	Expands all the groups in the tasks list.
Collapse All	Collapses all the groups in the tasks list.

All Descendants	Displays all tasks associated with the folder selected from the folder hierarchy and all of its child folders. When this option is not selected, the upper pane displays only those tasks associated with the selected folder.
Mark as Read	Displays the selected tasks in regular font (not bold), which indicates that they have been read.
Mark as Unread	Displays the selected tasks in bold font, which indicates that they have not been read.
Mark Thread as Read	Displays the entire thread in regular type, which indicates that the tasks it contains have been read.
Mark Thread as Unread	Displays the entire thread in boldface type, which indicates that the tasks it contains have not been read.
Flag	Flags an item. Only you can see and remove your flags. When an item is flagged, Yes displays in the Flag column in the upper pane. Note that this column does not display by default. Choose Task ▶ Filters ▶ Show Fields to add the column to the upper pane.
Remove Flag	Removes the flag from an item.
Delete	Deletes the task from the StarTeam repository.
Find	Finds all tasks containing text that matches the specified text string.
Find Next	Finds the next task containing text that matches the specified text string.
Find Previous	Finds the previous task containing text that matches the specified text string.
Filters ▶ Show Fields	Opens the Show Fields dialog box where you can select which column headers to display in the upper pane.
Filters ▶ Sort and Group	Opens the Sort and Group dialog box where you can specify how to sort and group the data in the upper pane based on up to four task properties.
Filters ▶ Queries	Opens the Queries dialog box where you can create or apply a query. Items that match the query become the rows displayed in the upper pane.
Filters ▶ Save Current Settings	Opens the Save Current Settings dialog box where you can save the current column headers, query, and method of sorting and grouping as a filter, which can be reapplied in the future.
Filters ▶ Reset Current Settings	Returns to the current default filter as defined on the server.
Filters ▶ Filters	Opens the Filters dialog box where you can display the filters available for this component. This option allows you to select a filter or create a new one. An asterisk preceding a filter name in the filter list box indicates that you have changed the contents of the upper pane since the filter was applied. For example, you may have displayed additional fields, grouped the items differently, or applied another query.
Reports	Opens the Reports dialog box where you can create a report using the selected task.
Charts ▶ Simple	Creates a simple chart. A simple chart contains only one series.
Charts ▶ Distribution	Creates a distribution chart. A distribution chart is the same as a pie chart. Each wedge indicates what fraction of the whole a group represents.
Charts ▶ Correlation	Creates a correlation chart. A correlation chart shows the relationship or degree of relationship between numeric values in several series.
Charts ▶ Time-series	Creates a time-series chart. A time-series chart is a line chart that shows the number of items that have the same day, week, or month in the specified time/ date field.
Links ▶ Create Link	Start a link from a selected task to another item.

Links ▶ Complete Link	Complete the link from a task to the selected item.
Links ▶ Cancel Link	Stop the process of creating a link.
Set Active Process Item ▶ Current View	Designates the selected task in the current view as the active process item. Active process items are used when you add tasks to a view or check them in.
Set Active Process Item ▶ Select View	Designates the selected task in an alternate view as the active process item. Active process items are used when you add tasks to a view or check them in.
Clear Active Process Item	Changes the selected task from active to inactive. Active process items are used when you add tasks to a view or check them in.

Related Procedures

[Using Tasks](#)

Related Reference

[Cross-Platform Client Menus](#)

Topic Menu

This topic describes the menu commands found on the **Topic** menu of the Cross-Platform Client.

The **Topic** menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

Topic Menu Command	Description
New	Creates a new topic.
Respond	Send a reply to a previous message.
Properties	Opens the Topic Properties dialog box for the selected topic.
Copy URL to Clipboard	Places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.
Send To	E-mail a copy of the topic properties to one or more team members.
Save Shortcut	Creates a shortcut to the selected topic. You can use the shortcut to start the application, open the project view in the configuration in which the shortcut was created, and open the Properties dialog box for the selected topic.
Lock/Unlock	Opens the Set My Lock Status dialog box where you can specify the lock status (unlocked, exclusive, or non-exclusive) for the folder or break an existing lock on a topic.
Compare Properties	Compares the properties of two selected topics.
Labels ▶ New	Opens the Attach a New Revision Label dialog box where you can create a new revision label and attach it to the selected topic or topics.
Labels ▶ Attach	Opens the Attach a Label dialog box where you can add an existing label to the selected topic or topics.
Labels ▶ Detach	Opens the Detach a Label dialog box where you can remove a label from the selected topic or topics.
Advanced ▶ Export	Opens the Export dialog box where you can export the data displayed in the upper pane for use in a spreadsheet, database, or another application.
Advanced ▶ Behavior	Opens the Topic Behavior dialog box where you can view or change the behavior of the selected topic.
Advanced ▶ Item Access Rights	Opens the Topic Access Rights dialog box for the selected topic.
Advanced ▶ Component Access Rights	Opens the Topic Component Access Rights dialog box.
Advanced ▶ Customize	Opens the Customize dialog box where you can create custom fields that become topic properties.
Advanced ▶ View Compare/Merge	Opens the View Compare/Merge Wizard .
Select ▶ All	Selects all topics displayed in the upper pane.
Select ▶ By Query	Opens the Select Query dialog box where you can specify a query that selects all items displayed in the upper pane that match the specified query.
Select ▶ By Label	Opens the Select a Label dialog box where you can specify a label that selects all items displayed in the upper pane that have the specified label.
Expand All	Expands all the groups in the topics list.
Collapse All	Collapses all the groups in the topics list.

All Descendants	Displays all topics associated with the folder selected from the folder hierarchy and all of its child folders. When this option is not selected, the upper pane displays only those topics associated with the selected folder.
Mark as Read	Displays the selected topics in regular font (not bold), which indicates that they have been read.
Mark as Unread	Displays the selected topics in bold font, which indicates that they have not been read.
Mark Thread as Read	Displays the entire thread in regular type, which indicates that the topics it contains have been read.
Mark Thread as Unread	Displays the entire thread in boldface type, which indicates that the topics it contains have not been read.
Flag	Flags an item. Only you can see and remove your flags. When an item is flagged, Yes displays in the Flag column in the upper pane. Note that this column does not display by default. Choose Topic ▸ Filters ▸ Show Fields to add the column to the upper pane.
Remove Flag	Removes the flag from an item.
Delete	Deletes the topic from the StarTeam repository.
Find	Finds all topics containing text that matches the specified text string.
Find Next	Finds the next topic containing text that matches the specified text string.
Find Previous	Finds the previous topic containing text that matches the specified text string.
Filters ▸ Show Fields	Opens the Show Fields dialog box where you can select which column headers to display in the upper pane.
Filters ▸ Sort and Group	Opens the Sort and Group dialog box where you can specify how to sort and group the data in the upper pane based on up to four topic properties.
Filters ▸ Queries	Opens the Queries dialog box where you can create or apply a query. Items that match the query become the rows displayed in the upper pane.
Filters ▸ Save Current Settings	Opens the Save Current Settings dialog box where you can save the current column headers, query, and method of sorting and grouping as a filter, which can be reapplied in the future.
Filters ▸ Reset Current Settings	Returns to the current default filter as defined on the server.
Filters ▸ Filters	Opens the Filters dialog box where you can display the filters available for this component. This option allows you to select a filter or create a new one. An asterisk preceding a filter name in the filter list box indicates that you have changed the contents of the upper pane since the filter was applied. For example, you may have displayed additional fields, grouped the items differently, or applied another query.
Reports	Opens the Reports dialog box where you can create a report using the selected topic.
Charts ▸ Simple	Creates a simple chart. A simple chart contains only one series.
Charts ▸ Distribution	Creates a distribution chart. A distribution chart is the same as a pie chart. Each wedge indicates what fraction of the whole a group represents.
Charts ▸ Correlation	Creates a correlation chart. A correlation chart shows the relationship or degree of relationship between numeric values in several series.
Charts ▸ Time-series	Creates a time-series chart. A time-series chart is a line chart that shows the number of items that have the same day, week, or month in the specified time/ date field.
Links ▸ Create Link	Start a link from a selected topic to another item.

Links ▶ Complete Link	Complete the link from a topic to the selected item.
Links ▶ Cancel Link	Stop the process of creating a link.
Set Active Process Item ▶ Current View	Designates the selected topic in the current view as the active process item. Active process items are used when you add topics to a view or check them in.
Set Active Process Item ▶ Select View	Designates the selected topic in an alternate view as the active process item. Active process items are used when you add topics to a view or check them in.
Clear Active Process Item	Changes the selected topic from active to inactive. Active process items are used when you add topics to a view or check them in.

Related Procedures

[Using Topics](#)

Related Reference

[Cross-Platform Client Menus](#)

Audit Menu

This topic describes the menu commands found on the **Audit** menu of the Cross-Platform Client.

The **Audit** menu is always available on the main window, regardless of the component selected from the upper pane. Depending upon the privileges assigned to you and the activities you need to perform, some options may not be enabled.

Audit Menu Command	Description
Copy URL to Clipboard	Places in the clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items. From the clipboard, you can paste the URL to a selected application.
Send To	E-mail a copy of an audit entry to one or more team members.
Export	Opens the Export dialog box where you can export the data displayed in the upper pane for use in a spreadsheet, database, or another application.
Select ▸ All	Selects all audits displayed in the upper pane.
Select ▸ By Query	Opens the Select Query dialog box where you can specify a query that selects all items displayed in the upper pane that match the specified query.
Expand All	Expands all the groups in the audits list.
Collapse All	Collapses all the groups in the audits list.
All Descendants	Displays all audits associated with the folder selected from the folder hierarchy and all of its child folders. When this option is not selected, the upper pane displays only those audits associated with the selected folder.
Find	Finds all audits containing text that matches the specified text string.
Find Next	Finds the next audit containing text that matches the specified text string.
Find Previous	Finds the previous audit containing text that matches the specified text string.
Filters ▸ Show Fields	Opens the Show Fields dialog box where you can select which column headers to display in the upper pane.
Filters ▸ Sort and Group	Opens the Sort and Group dialog box where you can specify how to sort and group the data in the upper pane based on up to four audit properties.
Filters ▸ Queries	Opens the Queries dialog box where you can create or apply a query. Items that match the query become the rows displayed in the upper pane.
Filters ▸ Save Current Settings	Opens the Save Current Settings dialog box where you can save the current column headers, query, and method of sorting and grouping as a filter, which can be reapplied in the future.
Filters ▸ Reset Current Settings	Returns to the current default filter as defined on the server.
Filters ▸ Filters	Opens the Filters dialog box where you can display the filters available for this component. This option allows you to select a filter or create a new one. An asterisk preceding a filter name in the filter list box indicates that you have changed the contents of the upper pane since the filter was applied. For example, you may have displayed additional fields, grouped the items differently, or applied another query.
Reports	Opens the Reports dialog box where you can create a report using the selected audit.
Charts ▸ Distribution	Creates a distribution chart. A distribution chart is the same as a pie chart. Each wedge indicates what fraction of the whole a group represents.
Charts ▸ Time-series	Creates a time-series chart. A time-series chart is a line chart that shows the number of items that have the same day, week, or month in the specified time/date field.

Related Procedures

[Viewing the Audit Log](#)

Related Reference

[Cross-Platform Client Menus](#)

Tools Menu

This topic describes the menu commands found on the **Tools** menu of the Cross-Platform Client.

Tools Menu Command	Description
Personal Options	Opens the Personal Options dialog box where you can customize the application.
My Account	Opens the My Account dialog box where you can edit your account information, change your password, or review the logged-on user's group membership information.
StarTeam Log	Displays the latest information from the client log, primarily for diagnostic purposes. The log can display error messages that occur during use of the application, summaries of each operation performed, details about each operation performed, and StarTeamMPX events.
Convert to Native Format	Converts files saved in the Native-I vault structure to the Native-II vault structure. Native-I vaults were last available in StarTeam 2005 Release 2.
File Annotation	Shows historical information about changes made to text files.
File Compare	Opens the embedded version of File Compare/Merge.

Related Procedures

[Configuring Your Client](#)

Related Reference

[Cross-Platform Client Menus](#)

Window Menu

This topic describes the menu commands found on the [Window](#) menu of the Cross-Platform Client.

Window Menu Command	Description
Cascade	Arranges open windows so that they overlap but all their titles are visible.
Tile Horizontally	Arranges open windows horizontally without overlapping them.
Tile Vertically	Arranges the open windows vertically without overlapping them.
Arrange Icons	Orders the icons for minimized windows.
Refresh	Refreshes the contents of the project window.
Component Refresh	Refreshes the contents of the upper pane without collapsing open groups of items. For example, if the File tab has been selected, this option refreshes the File list.
Component Refresh/Collapse	Refreshes the contents of the upper pane and simultaneously collapse open groups of items.
Foreign Refresh	Refreshes third-party projects or archives, such as Microsoft Visual SourceSafe projects and PVCS archive files.
Recent Views	Displays a list of the most recently accessed project view windows. A checkmark displays in front of the currently displayed project view. To display a different project view, click its name.

Related Reference

[Cross-Platform Client Menus](#)

Help Menu

This topic describes the menu commands found on the [Help](#) menu of the Cross-Platform Client.

Help Menu Command	Description
Help Topics	Opens the help system for StarTeam. Review help topics on the application's features. Allows you to view a Table of Contents or search for specific concepts in the Index.
About	Displays release and copyright information, and provides information about the Java virtual machine and the location of certain classes on your workstation.

Related Concepts

[Help on Help](#)

Related Reference

[Cross-Platform Client Menus](#)

Customization

This section contains reference topics related to customization.

In This Section

[Email Customization Reference](#)

This section contains reference topics related to the configuration and message templates used to configure custom email notifications in StarTeam.

[Detail Pane Customization Reference](#)

This section contains sample templates that you can use as a starting point for modifying the Detail (lower) pane in the Cross-Platform Client.

Email Customization Reference

This section contains reference topics for the configuration and message templates used to configure custom email notifications in StarTeam.

In This Section

[Change Request Configuration File](#)

Describes the `ChangeRequest.xml` configuration file used for customizing email notification messages.

[Change Request Message Template Syntax](#)

Describes the message template syntax used in the `cr-new.txt` file for customized email notifications.

Change Request Configuration File

This topic describes the `ChangeRequest.xml` configuration file. Your server configuration repository contains a *Notifications* subfolder containing this and other configuration files.

ChangeRequest.xml

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2. <notification-config version="1.0">
3.   <rule-list>
4.     <rule project="*" event="new" template="cr-new-html"/>
5.     <rule project="*" event="modified" template="cr-modified-html"/>
6.     <rule project="*Project*" event="new" template="cr-new-txt"/>
7.     <rule project="*Project*" event="modified" template="cr-modified-txt"/>
```

Note: `template` is a mandatory `rule` attribute. It specifies the message template that you wish to use for the notification and it must correspond to one of the template nodes, such as `<template id="cr-new-html">`. `event` and `project` are optional `rule` attributes. `event` specifies whether the item triggering the notification is being created ("new") or edited ("modified") or either of the two ("*"). If omitting the `event` attribute, the notification applies to any event. `project` allows the notification to be limited to a certain project or projects only. If omitted, the notification applies to all projects.

```
8.   </rule-list>
9.   <template-list>
10.    <template id="cr-new-html">
11.      <subject>New Change Request #~~ChangeNumber~~</subject>
12.      <body content-type="text/html" template-file=".\\cr-new.html" />
```

Note: You can give the `template-file` attribute an absolute file path or a path relative to the *Notifications* folder.

```
13.    </template>
14.    <template id="cr-modified-html">
15.      <subject>Modified Change Request #~~ChangeNumber~~</subject>
16.      <body content-type="text/html" template-file=".\\cr-modified.html" />
17.    </template>
18.    <template id="cr-new-txt">
19.      <subject>New Change Request #~~ChangeNumber~~</subject>
20.      <body content-type="text/plain" template-file=".\\cr-new.txt" />
21.    </template>
22.    <template id="cr-modified-txt">
23.      <subject>Modified Change Request #~~ChangeNumber~~</subject>
24.      <body content-type="text/plain" template-file=".\\cr-modified.txt" />
```

```
25. </template>
26. </template-list>
26. </notification-config>
```

Examining the Configuration File

In the file shown above, the `<rule-list>` element contains the following section of code:

```
<rule project="*" event="new" template="cr-new-html"/>
<rule project="*" event="modified" template="cr-modified-html"/>
<rule project="*Project*" event="new" template="cr-new-txt"/>
<rule project="*Project*" event="modified" template="cr-modified-txt"/>
```

When you create or modify a change request, the server searches through the rules defined in the `<rule-list>` element for the first project attribute that matches the name of the current project that consists of more than the asterisk wildcard character. This allows you to use `project="*"` for any projects where you have not created a more specific rule. For example, using the sample code shown above, a project names *Project2* would use the text templates and a project named *Whitestar* would use the HTML templates.

Next, the server searches the matching rule to determine if it applies to the current "new" or "modified" change request. The `event` attribute of the rule determines this by the values "new" or "modified". If this value is absent or set to "*", all change requests use the rule. If the current change request is new, a rule with an `event` attribute of "new" or "*" must be found.

Once the server finds the correct rule, the `template` attribute indicates where to look in this .xml file for the subject line to use in the email, the type of template file, and the path to the template file. In the sample *ChangeRequest.xml* file, if the `template` attribute in the selected rule was "cr-new-txt" the server uses the following template section to obtain more information:

```
<template id="cr-new-txt">
<subject>New Change Request #~~ChangeNumber~~</subject>
<body content-type="text/plain" template-file=".\\cr-new.txt" />
</template>
```

The subject line of the email reads *New Change Request* followed by the change request number. The template content type is "text/plain" and the template file is located at ".\\cr-new.txt". You can use the absolute path or a path relative to the *Notifications* folder for the location of the template file. This example uses a relative path to locate the template file within the *Notifications* folder.

Related Reference

[Fields](#)

Change Request Message Template Syntax

This topic describes the `cr-new.txt` message template file and examines message template syntax in general. You can use HTML or text format for your custom templates. Your server configuration repository contains a *Notifications* subfolder containing this file and other message template files in both HTML and text format for change requests, requirements, tasks, and topics.

cr-new.txt

This message has been sent to you automatically by StarTeam Server
because the Change Request described below has been created
by `~~ModifiedUserID~~`.

The Change Request is located in
Project: `~~project~~`
View: `~~view~~`

Property Summary:
Type: `~~type~~`
Status: `~~status~~`
Responsibility: `~~responsibility~~`
Priority: `~~priority~~`
Severity: `~~severity~~`
Platform: `~~platform~~`
Entered By: `~~EnteredBy~~`
Entered On: `~~EnteredOn~~`
Synopsis: `~~synopsis~~`
Description: `~~Description~~`

You can access Change Request #`~~ChangeNumber~~` by following this URL:
`~~url~~`

Examining the Message Template Syntax

Whenever you want to use the current value for a property field, enter the internal name for that field preceded and followed by two tildes (`~~`). For example, `Status: ~~status~~` displays the word “Status:” in the email followed by the current value of the status property.

The file also uses `~~url~~` to include the StarTeam URL for the change request.

Note: Client-calculated property fields cannot be used in the notification message templates. Refer to the Fields Reference link at the bottom of this topic for a list of all fields. The description for the field includes its internal name and whether it is a client-calculated property.

Displaying Values for Modified Fields

The text of `cr-modified.txt` uses the following syntax:

```
~~isnew.AddressedBy?Addressed By: ~~~isnew.AddressedBy~~~isnew.AddressedBy?~~
```

In these expressions, the `isnew` prefix precedes the internal name of the property field. This allows a template to specify that the property field value and/or additional formatting text should be included in the resulting notification email only if the value of the field has been changed – that is, only if the previous version had a different value for this field.

The text `Addressed By:` (from the first expression, `~~isnew.AddressedBy?Addressed By: ~~`), the value of the `AddressedBy` property (from the second expression `~~isnew.AddressedBy~~`) and a trailing linefeed (from the last expression) display in the email notification message only if the value of the `AddressedBy` property has been changed from the previous revision of the change request.

If you are using an HTML message template instead of the text version, you could display the output of the expression in bold text (only if it was changed for this revision) as follows:

```
Addressed By: ~~~isnew.AddressedBy? <b>~~~~AddressedBy~~~~isnew.AddressedBy?</b>~~
```

The above example behaves as follows in an email notification:

- ◆ `Addressed By:` (Text or formatting always included)
- ◆ `~~isnew.AddressedBy?~~` (Text or formatting included only if the `AddressedBy` field changes in this revision)
- ◆ `~~AddressedBy~~` (Value always included)
- ◆ `~~isnew.AddressedBy?~~` (Text or formatting included only if the `AddressedBy` field changes in this revision)

Definition of Message Template Syntax

In general, you can enter the following information into the message template files:

Note: In the following examples, `propname` could be `type`, `status`, `responsibility`, and so on. Refer to the example `cr-new.txt` file at the beginning of this topic for more examples of property field names.

- ◆ `~~propname~~`: Replaced with the value of the specified property.
- ◆ `~~old.propname~~`: Replaced with the old value of the specified property.
- ◆ `~~new.propname~~`: Replaced with the new value of the specified property.
- ◆ `~~project~~`: Replaced with the name of the project containing the item. This is not available as a normal item property.
- ◆ `~~view~~`: Replaced with the name of the view containing the item.
- ◆ `~~url~~`: Replaced with a StarTeam URL of the item.
- ◆ `~~isnew.propname~~`: Replaced with the value of the `propname` property only if the property has a new value. Otherwise, this expression resolves to an empty string.
- ◆ `~~isnew.propname?text-string~~`: Replaced with the text-string text only if the `propname` property has a new value. Otherwise, this expression resolves to an empty string.

Related Reference

[Fields](#)

Detail Pane Customization Reference

This section contains sample templates that you can use as a starting point for modifying the Detail (lower) pane in the Cross-Platform Client.

In This Section

[Sample Folder Template](#)

Provides an example of a HTML-based template used to modify the Detail pane for the Folder component.

[Sample File Template](#)

Provides an example of a HTML-based template used to modify the Detail pane for the File component.

[Sample Change Request Template](#)

Provides an example of a HTML-based template used to modify the Detail pane for the Change Request component.

[Sample Task Template](#)

Provides an example of a HTML-based template used to modify the Detail pane for the Task component.

[Sample Topic Template](#)

Provides an example of a HTML-based template used to modify the Detail pane for the Topic component.

[Sample Requirement Template](#)

Provides an example of a HTML-based template used to modify the Detail pane for the Requirement component.

[Sample Change Package Template](#)

Provides an example of a HTML-based template used to modify the Detail pane for the change package component.

Sample Folder Template

You can use the following sample for customizing the Detail pane for the Folder component.

Sample Folder HTML Template

```
<html>
<head></head>
<body>
<table bgcolor=#aaabbbccc width=100%>
<tr>
<td align=center><b>~~Name~~<b></td>
</tr>
</table>
<table>
<tr>
<td align=left><b>Status:</b></td>
<td>~~Status~~</td>
</tr>
<tr>
<td align=left><b>Working folder:</b></td>
<td>~~LocalPath~~</td>
</tr>
<tr>
<td align=left><b>Project Folder Path:</b></td>
<td>~~Folder Path~~</td>
</tr>
</table>
<hr>
<b>Last modified by: </b>~~Author~~, ~~Date&Time~~<br>
<b>Comment:</b><i>~~Comment~~</i>
</body>
</html>
```

Fields Used in Detail Pane Templates

The fields used in the Detail pane HTML templates are recognized by the client when they are contained between double tilde ~~ characters. For example, `~~LocalPath~~` represents the path to your local working folder represented by the **Complete working folder path** field found in the **Name** tab of the **Folder Properties** dialog box.

Refer to the link at the bottom of this topic for more information about the fields that you can use in the Detail pane templates.

Note: You can use the fields listed in this sample template in other templates provided with the Cross-Platform Client such as, report templates and email notification templates.

Related Procedures

[Customizing the Detail Pane](#)

Related Reference

[Detail Pane Customization Reference Fields](#)

Sample File Template

You can use the following sample for customizing the Detail pane for the File component.

Sample File HTML Template

```
<html>
<head></head>
<body>
<table bgcolor=#aaabbbccc width=100%>
<tr>
<td align=center><b>~~Name~~<b></td>
</tr>
</table>
<table>
<tr>
<td align=left><b>Status:</b></td>
<td>~~Status~~</td>
</tr>
<tr>
<td align=left><b>Size:</b></td>
<td>~~FileSize~~</td>
</tr>
<tr>
<td align=left><b>Working folder:</b></td>
<td>~~Path~~</td>
</tr>
<tr>
<td align=left><b>Project Folder Path:</b></td>
<td>~~Folder Path~~</td>
</tr>
</table>
<hr>
<b>Last modified by: </b>~~Author~~, ~~Date&Time~~<br>
<b>Comment:</b><i>~~Comment~~</i>
</body>
</html>
```

Fields Used in Detail Pane Templates

The fields used in the Detail pane HTML templates are recognized by the client when they are contained between double tilde ~~ characters. For example: ~~Path~~ represents the path to your local working folder represented by the **Path** field found in the **Working File** tab of the **File Properties** dialog box. Refer to the link at the bottom of this topic for more information about the fields that you can use in the Detail pane templates.

Note: You can use the fields listed in this sample template in other templates provided with the Cross-Platform Client such as, report templates and email notification templates.

Related Procedures

[Customizing the Detail Pane](#)

Related Reference

[Detail Pane Customization Reference](#)

[Fields](#)

Sample Change Request Template

You can use the following sample for customizing the Detail pane for the Change Request component.

Sample Change Request HTML Template

```
<html>
<head></head>
<body>
<table width=100% border=1>
<tr bgcolor=#aabbcc>
<th>CR Number</th>
<th>Status</th>
<th>Priority</th>
<th>Type</th>
<th>Responsibility</th>
</tr><tr>
<td align=center>~~ChangeNumber~~</td>
<td align=center>~~Status~~</td>
<td align=center>~~Priority~~</td>
<td align=center>~~Type~~</td>
<td align=center>~~Responsibility~~</td>
</tr>
</table>
<p align=right
<b>Entered By</b>: ~~EnteredBy~~, ~~EnteredOn~~ </p>
<b>Synopsis</b>:<br> ~~Synopsis~~ <br><br>
<b>Description</b>:<br> ~~Description~~ <br><br>
<b>Work Around</b>:<br> ~~WorkAround~~ <br><br>
<b>Fix</b>: ~~Fix~~<br> <br><hr>
<i>Last modified by: ~~ModifiedUserID~~, ~~ModifiedTime~~</i><br>
<b>Number of attachments</b>: ~~AttachmentCount~~<br>
<!--
<b>Flag User List</b>: ~~FlagUserList~~<br>
<b>Version</b>: ~~RevisionNumber~~<br>
<b>Branch State</b>: ~~BranchState~~<br>
<b>Read Status User List</b>: ~~ReadStatusUserList~~<br>
<b>Branch On Change</b>: ~~BranchOnChange~~<br>
```

Attachment IDs: ~~AttachmentIDs~~

Closed On: ~~ClosedOn~~

Component: ~~Component~~

Parent ID: ~~ParentObjectID~~

Root Object ID: ~~RootObjectID~~

Created Time: ~~CreatedTime~~

Share State: ~~ShareState~~

CommentID: ~~CommentID~~

Folder: ~~Folder~~

Created By: ~~CreatedUserID~~

Deleted Time: ~~DeletedTime~~

Dot Notation ID: ~~DotNotationID~~

Parent Revision: ~~PathRevision~~

Last Build Tested: ~~LastBuildTested~~

Non-Exclusive Lockers: ~~NonExclusiveLockers~~

Short Comment: ~~ShortComment~~

Locked By: ~~ExclusiveLocker~~

Folder Path: ~~Folder Path~~

Object ID: ~~ID~~

Flag: ~~Flag~~

Platform: ~~Platform~~

Severity: ~~Severity~~

Read Only: ~~ReadOnly~~

My Lock: ~~MyLock~~

Configuration Time: ~~ConfigurationTime~~

Comment: ~~Comment~~

Revision Flags: ~~RevisionFlags~~

Parent Branch Revision: ~~ParentRevision~~

External Reference: ~~ExternalReference~~

Category: ~~Category~~

End Modified Time: ~~EndModifiedTime~~

New Revision Comment: ~~NewRevisionComment~~

Addressed In: ~~AddressedIn~~

Resolved On: ~~ResolvedOn~~

View: ~~ViewID~~

Addressed In View: ~~AddressedInView~~

Addressed By: ~~AddressedBy~~


```
<b>Verified On</b>: ~~VerifiedOn~~<br>
<b>Deleted By</b>: ~~DeletedUserID~~<br>
<b>Test Command</b>: ~~TestCommand~~<br>
<b>Attachment names</b>: ~~AttachmentNames~~<br>
<b>Dot Notation</b>: ~~DotNotation~~<br>
<b>Read Status</b>: ~~ReadStatus~~<br>
-->
</body>
</html>
```

Fields Used in Detail Pane Templates

The fields used in the Detail pane HTML templates are recognized by the client when they are contained between double tilde ~~ characters. For example: `~~Status~~` represents the Status field found in the **Change Request Properties** dialog box. Refer to the link at the bottom of this topic for more information about the fields that you can use in the Detail pane templates.

Note: You can use the fields listed in this sample template in other templates provided with the Cross-Platform Client such as, report templates and email notification templates.

Related Procedures

[Customizing the Detail Pane](#)

Related Reference

[Detail Pane Customization Reference Fields](#)

Sample Task Template

You can use the following sample for customizing the Detail pane for the Task component.

Sample Task HTML Template

```
<html>
<head></head>
<body>
<b><b>MS WBS Code:</b> ~~StTaskWBSCode~~ <br>
<b>Attention Notes:</b> ~~StTaskAttentionNotes~~ <br>
<b>Estimated Start:</b> ~~StTaskEstimatedStart~~ <br>
<b>My Lock:</b> ~~MyLock~~ <br>
<b>Folder Path:</b> ~~Folder Path~~ <br>
<b>Estimated Hours Variance:</b> ~~StTaskEstimatedHoursVariance~~ <br>
<b>Task Duration:</b> ~~StTaskDuration~~ <br>
<b>Version:</b> ~~RevisionNumber~~ <br>
<b>Resource IDs:</b> ~~StTaskResourceIDs~~ <br>
<b>Flag:</b> ~~Flag~~ <br>
<b>Short Comment:</b> ~~ShortComment~~ <br>
<b>Created By:</b> ~~CreatedUserID~~ <br>
<b>Responsibility:</b> ~~StTaskResponsibility~~ <br>
<b>Constraint Date:</b> ~~StTaskConstraintDate~~ <br>
<b>Created Time:</b> ~~CreatedTime~~ <br>
<b>Share State:</b> ~~ShareState~~ <br>
<b>Locked By:</b> ~~ExclusiveLocker~~ <br>
<b>Priority:</b> ~~StTaskPriority~~ <br>
<b>Resource Count:</b> ~~StTaskResourceCount~~ <br>
<b>Estimated Finish:</b> ~~StTaskEstimatedFinish~~ <br>
<b>Actual Start:</b> ~~StTaskActualStart~~ <br>
<b>Actual Finish:</b> ~~StTaskActualFinish~~ <br>
<b>Estimated Hours:</b> ~~StTaskEstimatedHours~~ <br>
<b>Is My Task?:</b> ~~StTaskIsMyTask~~ <br>
<b>Attachment IDs:</b> ~~AttachmentIDs~~ <br>
<b>Estimated Start Variance:</b> ~~StTaskEstimatedStartVariance~~ <br>
<b>Deleted By:</b> ~~DeletedUserID~~ <br>
<b>Dot Notation:</b> ~~DotNotation~~ <br>
<b>Parent Task ID:</b> ~~StTaskParentID~~ <br>
```

MS Task Unique ID: ~~StTaskUniqueID~~

 Status: ~~StTaskStatus~~

 Notes: ~~StTaskNotes~~

 Read Status: ~~ReadStatus~~

 Children Count: ~~ChildrenCount~~

 Constraint Type: ~~StTaskConstraintType~~

 Last Work/Dependency Update: ~~StWorkDependencyLastUpdate~~

 Modified By: ~~ModifiedUserID~~

 New Revision Comment: ~~NewRevisionComment~~

 Non-Exclusive Lockers: ~~NonExclusiveLockers~~

 Resource Names: ~~StTaskResourceNames~~

 Read Status User List: ~~ReadStatusUserList~~

 Task Type: ~~StTaskType~~

 Estimated Finish Variance: ~~StTaskEstimatedFinishVariance~~

 Actual Hours: ~~StTaskActualHours~~

 Revision Flags: ~~RevisionFlags~~

 Percent Complete: ~~StTaskPercentComplete~~

 Task Name: ~~StTaskName~~

 Attachment Count: ~~AttachmentCount~~

 CommentID: ~~CommentID~~

 Is Replicated: ~~Is Replicated~~

 Flag User List: ~~FlagUserList~~

 Object ID: ~~ID~~

 Task Origin: ~~StTaskOrigin~~

 Modified Time: ~~ModifiedTime~~

 MS Project File Name: ~~StTaskMSProjectFileName~~

 Deleted Time: ~~DeletedTime~~

 Milestone: ~~StTaskMilestone~~

 Work Record Count: ~~WorkRecCount~~

 Configuration Time: ~~ConfigurationTime~~

 Last MS Project Update: ~~StTaskMSProjectLastUpdate~~

 MS Task GUID: ~~StTaskGUID~~

 End Modified Time: ~~EndModifiedTime~~

 Task Number: ~~StTaskNumber~~

 Needs Attention: ~~StTaskNeedsAttention~~

 Attachment names: ~~AttachmentNames~~

 Comment: ~~Comment~~


```
<b>Read Only:</b> ~~ReadOnly~~ <br>
</body>
</html>
```

Fields Used in Detail Pane Templates

The fields used in the Detail pane HTML templates are recognized by the client when they are contained between double tilde ~~ characters. For example: `~~StTaskEstimatedStart~~` represents the estimated start date field found in the **Task Properties** dialog box. Refer to the link at the bottom of this topic for more information about the fields that you can use in the Detail pane templates.

Note: You can use the fields listed in this sample template in other templates provided with the Cross-Platform Client such as, report templates and email notification templates.

Related Procedures

[Customizing the Detail Pane](#)

Related Reference

[Detail Pane Customization Reference Fields](#)

Sample Topic Template

You can use the following sample for customizing the Detail pane for the Topic component.

Sample Topic HTML Template

```
<html>
<head></head>
<body>
<b>Flag User List</b>: ~~FlagUserList~~<br>
<b>Version</b>: ~~RevisionNumber~~<br>
<b>Status</b>: ~~Status~~<br>
<b>Read Status User List</b>: ~~ReadStatusUserList~~<br>
<b>Modified Time</b>: ~~ModifiedTime~~<br>
<b>Attachment IDs</b>: ~~AttachmentIDs~~<br>
<b>Created Time</b>: ~~CreatedTime~~<br>
<b>Content</b>: ~~Description~~<br>
<b>Share State</b>: ~~ShareState~~<br>
<b>CommentID</b>: ~~CommentID~~<br>
<b>Created By</b>: ~~CreatedUserID~~<br>
<b>Deleted Time</b>: ~~DeletedTime~~<br>
<b>Children Count</b>: ~~ChildrenCount~~<br>
<b>Title</b>: ~~Title~~<br>
<b>Non-Exclusive Lockers</b>: ~~NonExclusiveLockers~~<br>
<b>Topic Number</b>: ~~TopicNumber~~<br>
<b>Recipient IDs</b>: ~~RecipientIDs~~<br>
<b>Short Comment</b>: ~~ShortComment~~<br>
<b>Recipient Count</b>: ~~RecipientCount~~<br>
<b>Locked By</b>: ~~ExclusiveLocker~~<br>
<b>Folder Path</b>: ~~Folder Path~~<br>
<b>Object ID</b>: ~~ID~~<br>
<b>Flag</b>: ~~Flag~~<br>
<b>Recipient Names</b>: ~~RecipientNames~~<br>
<b>Read Only</b>: ~~ReadOnly~~<br>
<b>My Lock</b>: ~~MyLock~~<br>
<b>Configuration Time</b>: ~~ConfigurationTime~~<br>
<b>Comment</b>: ~~Comment~~<br>
<b>Revision Flags</b>: ~~RevisionFlags~~<br>
```

```
<b>End Modified Time</b>: ~~EndModifiedTime~~<br>
<b>Am I Recipient?</b>: ~~AmIRecipient~~<br>
<b>New Revision Comment</b>: ~~NewRevisionComment~~<br>
<b>Attachment Count</b>: ~~AttachmentCount~~<br>
<b>Type</b>: ~~Type~~<br>
<b>Priority</b>: ~~Priority~~<br>
<b>Modified By</b>: ~~ModifiedUserID~~<br>
<b>Deleted By</b>: ~~DeletedUserID~~<br>
<b>Attachment names</b>: ~~AttachmentNames~~<br>
<b>Dot Notation</b>: ~~DotNotation~~<br>
<b>Read Status</b>: ~~ReadStatus~~<br>
<b>Parent Topic ID</b>: ~~ParentTopicID~~<br>
</body>
</html>
```

Fields Used in Detail Pane Templates

The fields used in the Detail pane HTML templates are recognized by the client when they are contained between double tilde ~~ characters. For example: ~~Status~~ represents the status field found in the **Topic Properties** dialog box. Refer to the link at the bottom of this topic for more information about the fields that you can use in the Detail pane templates.

Note: You can use the fields listed in this sample template in other templates provided with the Cross-Platform Client such as, report templates and email notification templates.

Related Procedures

[Customizing the Detail Pane](#)

Related Reference

[Detail Pane Customization Reference Fields](#)

Sample Requirement Template

You can use the following sample for customizing the Detail pane for the Requirement component.

Sample Requirement HTML Template

```
<html>
<head></head>
<body>
<b>Requirement</b>: #~~Number~~<br>
<b>Name</b>: ~~Name~~<br>
<b>Version</b>: ~~RevisionNumber~~<br>
<b>Modified By</b>: ~~ModifiedUserID~~<br>
<b>Modified On</b>: ~~ModifiedTime~~<br>
<b>Comment</b>: ~~Comment~~<br>
<b>Created By</b>: ~~CreatedUserID~~<br>
<b>Created On</b>: ~~CreatedTime~~<br>
<b>Owner</b>: ~~Owner~~<br>
<b>Status</b>: ~~Status~~<br>
<b>Priority</b>: ~~Priority~~<br>
<b>Locked By</b>: ~~ExclusiveLocker~~<br>
<b>Description</b>: ~~Description~~<br>
<b>Attachments</b>: ~~AttachmentLinks~~<br>
<b>Attachment Count</b>: ~~AttachmentCount~~<br>
<b>Attachment names</b>: ~~AttachmentNames~~<br>
<b>Read Status User List</b>: ~~ReadStatusUserList~~<br>
<b>Share State</b>: ~~ShareState~~<br>
<b>CommentID</b>: ~~CommentID~~<br>
<b>Disabled</b>: ~~Disabled~~<br>
<b>Children Count</b>: ~~ChildrenCount~~<br>
<b>Child Type</b>: ~~ChildType~~<br>
<b>Non-Exclusive Lockers</b>: ~~NonExclusiveLockers~~<br>
<b>Short Comment</b>: ~~ShortComment~~<br>
<b>Recipient Count</b>: ~~RecipientCount~~<br>
<b>Folder Path</b>: ~~Folder Path~~<br>
<b>Object ID</b>: ~~ID~~<br>
<b>Flag</b>: ~~Flag~~<br>
<b>Read Only</b>: ~~ReadOnly~~<br>
```

```

<b>My Lock</b>: ~~MyLock~~<br>
<b>Configuration Time</b>: ~~ConfigurationTime~~<br>
<b>End Modified Time</b>: ~~EndModifiedTime~~<br>
<b>Am I Responsible?</b>: ~~AmIResponsible~~<br>
<b>New Revision Comment</b>: ~~NewRevisionComment~~<br>
<b>Type</b>: ~~Type~~<br>
<b>Attachment IDs</b>: ~~AttachmentIDs~~<br>
<b>Dot Notation</b>: ~~DotNotation~~<br>
<b>Read Status</b>: ~~ReadStatus~~<br>
<b>Parent Requirement ID</b>: ~~ParentRequirementID~~<br>
<b>Ambiguities Found</b>: ~~AmbiguitiesFound~~<br>
<b>Expected Effort</b>: ~~ExpectedEffort~~<br>
<b>External Reference</b>: ~~ExternalReference~~<br>
<b>High Effort</b>: ~~HighEffort~~<br>
<b>Low Effort</b>: ~~LowEffort~~<br>
<b>Notes</b>: ~~Notes~~<br>
<b>Responsible Count</b>: ~~ResponsibleCount~~<br>
<b>Responsible Names</b>: ~~ResponsibleNames~~<br>
<b>Revised Description</b>: ~~RevisedDescription~~<br>
</body>
</html>

```

Fields Used in Detail Pane Templates

The fields used in the Detail pane HTML templates are recognized by the client when they are contained between double tilde ~~ characters. For example: `~~Status~~` represents the Status field found in the **Requirement** dialog box. Refer to the link at the bottom of this topic for more information about the fields that you can use in the Detail pane templates.

Note: You can use the fields listed in this sample template in other templates provided with the Cross-Platform Client such as, report templates and email notification templates.

Related Procedures

[Customizing the Detail Pane](#)

Related Reference

[Detail Pane Customization Reference Fields](#)

Sample Change Package Template

You can use the following sample for customizing the Detail pane for the change package component.

Sample Change Package HTML Template

```
<html>
<head></head>
<body>
<b>Change package</b>: ~~Name~~<br>
<b>Change package</b>: ~~State~~<br>
<b>Change package</b>: ~~Revision~~<br>
<b>Change package</b>: ~~DotNotation~~<br>
<b>Created By</b>: ~~CreatedUserID~~<br>
<b>Created Time</b>: ~~CreatedTime~~<br>
<b>Last Modified</b>: ~~ModifiedUserID~~<br>
<b>Last Modified</b>: ~~ModifiedTime~~<br>
<b>Committed</b>: ~~CommitUserID~~<br>
<b>committed</b>: ~~CommitTime~~<br>
<b>Description</b>: ~~Description~~<br>
<b>Session Type</b>: ~~SessionType~~<br>
<b>Responsibility</b>: ~~Responsibility~~<br>
<b>Source view</b>: ~~SourceViewID~~<br>
<b>Target view</b>: ~~TargetViewID~~<br>
<b>Transaction ID</b>: ~~TransactionID~~<br>
<b>Comment</b>: ~~Comment~~<br>
<b>Committed in build</b>: ~~CommittedInBuild~~<br>
<b>Attachments</b>: ~~AttachmentLinks~~<br>
<b>Pre-commit View label</b>: ~~PreCommitViewLabel~~<br>
<b>Post-commit View label</b>: ~~PostCommitViewLabel~~<br>
<b>Pre-commit Revision label</b>: ~~PreCommitRevisionLabel~~<br>
<b>Post-commit Revision label</b>: ~~PostCommitRevisionLabel~~<br>
<b>Locked By</b>: ~~ExclusiveLocker~~<br>
<b>Non-Exclusive lockers</b>: ~~NonExclusiveLockers~~<br>
<b>Read Only</b>: ~~ReadOnly~~<br>
</body>
</html>
```


Fields Used in Detail Pane Templates

The fields used in the Detail pane HTML templates are recognized by the client when they are contained between double tilde ~~ characters. For example: ~~Status~~ represents the status field found in the **Topic Properties** dialog box. Refer to the link at the bottom of this topic for more information about the fields that you can use in the Detail pane templates.

Note: You can use the fields listed in this sample template in other templates provided with the Cross-Platform Client such as, report templates and email notification templates.

Related Procedures

[Customizing the Detail Pane](#)

Related Reference

[Detail Pane Customization Reference
Fields](#)

Keyboard Shortcuts

This section contains tables of keyboard shortcuts.

In This Section

[Client Keyboard Shortcuts](#)

Table of client keyboard shortcuts

[File Compare/Merge Keyboard Shortcuts](#)

Presents the keyboard shortcuts for use in the File Compare/Merge .

[Server Administration Keyboard Shortcuts](#)

Table of keyboard shortcuts for the Server Administration window.

Client Keyboard Shortcuts

Below are the keyboard shortcuts for the client.

Action	Keyboard Shortcut
Cancel Link Creation	ESCAPE
Check In	CTRL+I
Check In and Unlock	CTRL+U
Check Out	CTRL+G
Check Out and Unlock	CTRL+O
Component Refresh	F5
Component Refresh/Collapse	CTRL+F5
Delete	DELETE
Find	CTRL+F
Find Next	F3
Find Previous	SHIFT+F3
Flag	CTRL+F2
Remove Flag	CTRL+SHIFT+F2
Foreign Refresh	F6
Launch File Annotation tool	ALT+F1
Launch File Compare tool	ALT+F2
Launch Help	F1
Lock/Unlock	CTRL+L
Project New	CTRL+N
Project Open	CTRL+P
Redo character in Text Edit	CTRL+Y
Select All	CTRL+A
Show Item Properties	ALT+ENTER
Undo character in Text Edit	CTRL+Z
Window Refresh	SHIFT+F5

Note: On the Linux platform, ALT+F1 and ALT+F2 are designated as KDE global shortcuts. In order to use these shortcuts to launch the File Annotation and File Compare tools in the client, you must first disable them in the KDE application.

Related Reference

[File Compare/Merge Keyboard Shortcuts](#)
[Server Administration Keyboard Shortcuts](#)

File Compare/Merge Keyboard Shortcuts

This table presents the keyboard shortcuts for use in File Compare/Merge.

Action	Keyboard Shortcut
New File Comparison	CTRL+N
New Folder Comparison	CTRL+SHIFT+N
New Folder Merge	CTRL+SHIFT+M
New Image Comparison	CTRL+ALT+N
Save All	CTRL+SHIFT+S
Print	CTRL+P
Undo	CTRL+Z
Redo	CTRL+Y
Cut	CTRL+X
Copy	CTRL+C
Paste	CTRL+V
Find	CTRL+F
Find Next	F3
Find Previous	SHIFT+F3
Go To Line	CTRL+G
Previous Change	F7 OR ALT+UP
Next Change	F8 OR ALT+DOWN
Next Tab	F6
Previous Tab	SHIFT+F6
Maximize/Minimize Pane	CTRL+M
Center Splitter	CTRL+H
Move Splitter	CTRL+SHIFT+S
Reload	F5
Open Options Dialog Box	CTRL+O
Replace Text Left To Right (2-way merge)	ALT+RIGHT
Replace Text Right To Left (2-way merge)	ALT+LEFT
Append Text Left To Right (2-way merge)	ALT+SHIFT+RIGHT
Append Text Right To Left (2-way merge)	ALT+SHIFT+LEFT
Delete Text in Left Pane (2-way merge)	ALT+CTRL+LEFT
Delete Text in Right Pane (2-way merge)	ALT+CTRL+RIGHT
Move Text Up	ALT+SHIFT+UP
Move Text Down	ALT+SHIFT+DOWN
Replace Text Center To Left	ALT+<
Replace Text Center To Right	ALT+>

Replace Text Left To Center	ALT+M
Replace Text Right To Center	ALT+/
Append Text Left To Center	ALT+SHIFT+M
Append Text Right To Center	ALT+SHIFT+/
Append Text Center To Left	ALT+SHIFT+<
Append Text Center To Right	ALT+SHIFT+>

Related Concepts

[Overview of File Compare/Merge](#)
[File Compare/Merge UI](#)

Related Procedures

[Comparing and Merging Files](#)
[Setting File Compare/Merge Options](#)

Related Reference

[File Compare/Merge Options](#)
[File Compare/Merge Actions](#)

Server Administration Keyboard Shortcuts

Below are the keyboard shortcuts for the Server Administration window.

Action	Keyboard Shortcut
New Group	CTRL+G
New User	CTRL+G

Related Reference

[Client Keyboard Shortcuts](#)

[File Compare/Merge Keyboard Shortcuts](#)

Project, Folder, and Item Properties

This section contains detailed information about the properties for all types of items.

In This Section

[Folder Properties](#)

Lists and defines the folder properties.

[File Properties](#)

Lists and defines the file properties.

[Change Request Properties](#)

Lists and defines the change request properties.

[Requirement Properties](#)

Lists and defines the requirement properties.

[Task Properties](#)

Lists and defines the task properties.

[Topic Properties](#)

Lists and defines the topic properties.

[Project Properties Dialog Box](#)

Describes the project properties dialog box options.

Folder Properties

This topic presents the folder properties and their descriptions as displayed in the **Folder Properties** dialog box. The **Folder Properties** dialog box contains the following tabbed pages of properties.

Name

The following properties are on the **Name** page.

Property	Description
Name	Displays the name of the file.
Description	Displays the file description.
Created By	Displays the name of the person who created the file.
(Created) On	Displays the date on which the file was created.
Visible	Indicates whether the file is to be visible in the view or not.
Working Folder	<p>Displays the path to the default and alternate working folders.</p> <p>Default: This path location points to the default working folder, and applies to everyone accessing the project repository. DO NOT CHANGE this path unless you are a project administrator.</p> <p>Alternate: This path points to an alternate working folder on your machine. Specifying an alternate working folder affects only you, not any of the other team members.</p>
Complete Working Folder Path	Displays the complete path to the selected working folder (default or alternate.)

Exclude

The following properties are on the **Exclude** page.

Property	Description
Files To Be Excluded	<p>Indicates which files or types of files to exclude from visibility in the folder. The exclude list has no effect on files that are already part of the project. It only affects those with Not In View as their status. Exclude lists can be inherited from parent folders. Exclude options include:</p> <p>Inherit And Use Local Exclude List: Indicates that files matching the exclude list specifications set for this folder and those of its parent folder will be excluded.</p> <p>Use Local Exclude List: Indicates that files matching the only specifications set for the exclude list of this folder will be excluded.</p> <p>No Exclude List: Specifies that all files are included in the folder.</p>
Local Exclude List	<p>Displays the exclude specifications to use for excluding files from this folder. The exclude list is limited to a maximum of 255 characters. It contains file specifications (using the standard * and ? wild cards), separated by commas, spaces, or semicolons. To include a comma, space, or semicolon as part of the specification, enclose the specification in double-quotes. For example,</p> <pre>*.exe,*.dll p*z.doc;*.t?t "test *.*"</pre>
Inherited Exclude List	Displays the exclude specifications to use from the parent folder for excluding files from this folder.

History

The **History** page displays all the revisions of the folder. The following properties are displayed on the **History** page for each revision.

Property	Description
View	Displays the name of the view to which this folder belongs.
Revision	Displays the file revision number.
Modified By	Displays the name of the person who created the folder.
Modified Time	Displays the date and time the revision created.
Comment	Displays a comment explaining why the revision was created.
Dot Notation	Displays the branch number of the revision.

Link

The following properties are on the **Link** page which displays all the links to this folder.

Property	Description
Created By	Displays the name of the person who created the link to the folder.
Created On	Displays the date on which the link was created.
View	Displays the name of the current view if the link was created in the current view, or displays the name of view where the link was created and from which the link is shared.
Folder	Displays the name of the folder in which the folder or item in the link resides.
Item Type	Identifies the type of item to which the target end of the link is attached. This item is listed in the link list.
Item	Identifies the item to which the target end of the link is attached. It is identified by its folder name, file name, change request number, task number, topic number, or requirement number.
Item Details	Describes the item, using a folder description, file description, change request synopsis, task name, topic title, or requirement name.
Item Version	Displays the version number of the target end of the link if that revision is in the current view. When no revision number is displayed in the column, that end of the link is floating rather than pinned.
Selection Version	Displays the version number of the source end of the link if that revision is in the current view. When no revision number is displayed in the column, that end of the link is floating rather than pinned.
Comment	Displays a comment about this particular link.
File Status	Displays the status of a file that is linked to the folder.
Locked By	Displays the name of the person who locked the file linked to the folder.
Folder Path	Shows folder path information only when the linked item is in the same view. Otherwise, it displays the message, "Unavailable. Item in another view."

Related Procedures

[Viewing or Modifying Item Properties](#)

Related Reference

[Change Request Properties](#)

[File Properties](#)

[Requirement Properties](#)

[Task Properties](#)

[Topic Properties](#)

File Properties

This topic presents the file properties and their descriptions as displayed in the **File Properties** dialog box. The **File Properties** dialog box contains the following tabbed pages of properties.

General

The following properties are on the **General** page.

Property	Description
Name	Displays the name of the file.
Description	Displays the description of the file.
Status	Indicates the relationship between the copy of the file in your working folder and the tip revision in the repository.
Size	Displays the size of the tip revision of the file in bytes.
Last Modified By	Displays the name of the person who last modified the file.
Last Modified On	Displays the date on which the file was last modified.
File Time Stamp	Displays the time at which the file was last modified.
Locked Exclusively By	Displays the name of the user who has exclusively locked the file.
Locked Non-Exclusively By	The name of user who has non-exclusively locked a file.
EOL check-out format:	Displays the options for EOL formatting for text files during check-out: Client Defined: Causes workstation default or per-checkout EOL conversion option to be used. Fixed CR, Fixed LF, and Fixed CRLF: Causes this EOL format to be used always; the workstation/check-out conversion option is ignored. .
File type:	Indicates which file type is selected: ASCII, Binary, or Unicode.
Executable	Indicates whether the executable bit should be set for a UNIX file.

Working File

The following properties are on the **Working File** page.

Property	Description
Path	Displays the Path to the working folder for the file.
File Exists	Indicates whether the file exists in the working folder.
Size	Displays size of the file in the working folder in bytes.
Time Stamp	Displays the time that the working folder file was last modified.
Executable	Indicates whether the executable bit should be set for a UNIX file.

Archive

The following properties are on the **Archive** page.

Property	Description
Type	Indicates whether a file is stored as a StarTeam (Native) file, or as another type of file.
Format	Indicates the vault storage format for the file. StarTeam 2006 uses only Native-II storage.
Archive File Name	Displays the name of the Native-II Vault file that stores the tip revision. This name is the MD5 value of that file revision's content, converted to a 32-digit hex string.
Compression	<p>Indicates a file's level of compression.</p> <p>Default: A compromise between Maximize Compression and Maximize Speed</p> <p>Maximize Compression: The densest possible compression of file revisions. Used to save space on the server.</p> <p>Maximize Speed: The fastest possible compression of file revisions. Used to improve server performance.</p> <p>None: No compression.</p>

Custom

You can create custom properties for an item which will display in the item **Properties** dialog box.

The following properties are on the **Custom** page.

Property	Description
Property	Displays each custom property name.
Value	Displays the values for each custom property. Double-click the property name to edit the value.

Comment

The following properties are on the **Comment** page.

Property	Description
Comment For This Revision	Displays the reason for the changes to the current revision.
Comment For New Revision	Displays the reason for the changes to the new revision.

Related Procedures

[Viewing or Modifying Item Properties](#)

Related Reference

[Change Request Properties](#)

[Folder Properties](#)

[Requirement Properties](#)

[Task Properties](#)

[Topic Properties](#)

Change Request Properties

This topic presents a the change request properties and their descriptions as displayed in the **Change Request Properties** dialog box. The **Change Request Properties** dialog box contains the following tabbed pages of properties.

Synopsis

The following properties are on the **Synopsis** page.

Property	Description
Status	Displays the status of the change request.
Priority	Displays the priority level of the change request. Many people use repository customization to extend this field to include other values because Boolean values in the application are treated as enumerated types. For example, No is 0 and Yes is 1. An administrator might change No to Not A Priority, Yes to Priority 1, and add Priorities 2 through 10.
Type	Displays the type of change request, a Defect or a Suggestion .
Severity	Indicates the severity of the change request: Low , Medium , or High .
Platform	Indicates which operating system platform the to which the change request applies.
Last Build Tested	Displays the build label selected by a user to represent the last build in which a change request was tested.
External Reference	Indicates the customer or other outside source who provided the data for this change request.
Addressed In Build	Indicates the next build label created and applied to the view after the resolution to a change request occurs.
Component	Displays the component in which the defect occurs. It is often used with the Category property to narrow that identification to a subcomponent.
Category	Displays the name of the subcomponent in which the defect occurs. It is usually used in combination with the Component property.
Synopsis	Displays a brief description of the change request.
Responsibility	Displays the name of the person currently responsible for the change request.
Entered By	Displays the name of the person who entered the change request.

Description

The following properties are on the **Description** page. This page also contains **Browse** button for locating the command to test, and a **Run** button for running the test.

Property	Description
Description And Steps To Reproduce	Displays a complete, detailed description of the change request.
Test Command	Displays the command to use to test the solution for the change request.

Solution

The following properties are on the **Solution** page.

Property	Description
Work Around	Explains the solution to the change request other than the fix.
Fix	Displays the solution to the problem addressed by the change request.

Custom

You can create custom properties for an item which will display in the item **Properties** dialog box. The following properties are on the **Custom** page.

Property	Description
Property	Displays each custom property name.
Value	Displays the values for each custom property. Double-click the property name to edit the value.

Attachments

The **Attachments** page contains a list of all the files attached to the current change request.

Comment

The following properties are on the **Comment** page.

Property	Description
Comment For This Revision	Displays the reason for the changes to the current revision.
Comment For New Revision	Displays the reason for the changes to the new revision.

Related Procedures

[Viewing or Modifying Item Properties](#)

Related Reference

[File Properties](#)
[Folder Properties](#)
[Requirement Properties](#)
[Task Properties](#)
[Topic Properties](#)

Requirement Properties

This topic presents the requirement properties and their descriptions as displayed in the **Requirement Properties** dialog box. The **Requirement Properties** dialog box contains the following tabbed pages of properties.

Requirement

The following properties are on the **Requirement** page.

Property	Description
Name	Displays the requirement name.
Created By	Displays the name of person who created the first revision of the requirement in the view.
Created On	Displays the date on which first revision of the requirement was created.
Attachments	Indicates the number of files attached to the requirement.
Modified By	Displays the name of the last person who last modified the requirement.
Modified On	Displays the date on which the requirement was last modified.
Type	Displays the requirement type.
Owner	Displays the name of person ultimately responsible for the fulfillment of the requirement.
Status	Displays the current status of the requirement. This indicates the progress from submitted to rejected or completed. Note: The status ReadyForCCP means the requirement is ready for review by the Change Control Board.
External Reference	External source or reference for this requirement. This usually is the name of an external customer who asked for the requirement. If you are publishing requirements from CaliberRM to StarTeam, this property displays its identification for this requirement.
Description	Provides a description of the requirement, usually revised over time to eliminate ambiguities.

Responsibility

The **Responsibility** page lists the people responsible for completion of the requirement. You can add or remove people from the list.

These people will be notified of changes to the requirement if notification is enabled.

Ambiguity Review

The following properties are on the **Ambiguity Review** page. Reviewers will use the **Ambiguity Review** page to locate ambiguities in the initial description and revise that description.

Property	Description
Number Of Ambiguities Found	Indicates the number of ambiguities reviewers have found in the initial description of the requirement.
Revised Description	Provides a new, revised description because of ambiguities found in the original description or for other reasons.
Comments	Provides comments stating what the ambiguities are in the original requirement and why you have made the changes to the description.

Estimate

The following properties are on the **Estimate** page.

Property	Description
Expected Effort	Indicates the expected case estimate for how long it will take to implement the requirement fully. If you are publishing requirements from CaliberRM to StarTeam, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Low Effort and the High Effort fields, and should be used consistently for all requirements.
High Effort	Indicates the worst case estimate for how long it will take to implement the requirement fully. If you are publishing requirements from CaliberRM to StarTeam, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Low Effort and the Expected Effort properties, and should be used consistently for all requirements.
Low Effort	Indicates the best case estimate for how long it will take to implement the requirement fully. If you are importing requirements from CaliberRM, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Expected Effort and the High Effort fields, and should be used consistently for all requirements.

Custom

You can create custom properties for an item which will display in the item **Properties** dialog box.

The following properties are on the **Custom** page.

Property	Description
Property	Displays each custom property name.
Value	Displays the values for each custom property. Double-click the property name to edit the value.

Attachments

The **Attachments** page contains a list of all the files attached to the current requirement.

Comment

The following properties are on the **Comment** page.

Property	Description
Comment For This Revision	Displays the reason for the changes to the current revision.
Comment For New Revision	Displays the reason for the changes to the new revision.

Related Procedures

[Viewing or Modifying Item Properties](#)

Related Reference

[Change Request Properties](#)

[File Properties](#)

[Folder Properties](#)

[Task Properties](#)

[Topic Properties](#)

Task Properties

This topic presents the task properties and their descriptions as displayed in the **Task Properties** dialog box. The **Task Properties** dialog box contains the following tabbed pages of properties.

Task

The following properties are on the **Task** page.

Property	Description
Subtask Of	Displays the name of task for which this item is a subtask (if this item is a subtask).
Name	Displays the name of task or subtask.
Responsibility	Displays the name of the person responsible for the completion of this task or subtask. Other people can be assigned as additional resources.
Milestone	Indicates that the task or subtask should be treated as a milestone.
Status	Displays the task status: Pending: Waiting for completion of a predecessor task. Ready To Start: Work can be started on the task. In Progress: Work has been entered for the task. Finished: Work is finished on the task. Closed: Task is completed and closed. Hold Work temporarily stopped on the task, usually to wait for completion of another task.
Priority	Displays the task priority level. The default is Medium . These priorities are identical to those in MS Project. Do Not Level is a Microsoft Project-specific term you should ignore.
Duration	Indicates the number of hours expected for completion of the task.
Percent Complete	Displays the percentage of work that has been completed on a task.
Needs Attention	Notifies team leaders or task reviewers that this task requires attention. Enter the information about why this task needs attention in the text box below the Needs Attention check box.

Resources

The **Resources** page lists the task resources. You can assign responsibility to team members by adding them to the list with the **Add** button, and you can remove them using the **Remove** button.

Time

The following properties are on the **Time** page.

Property	Description
Plan Start	Displays the start date for the task.
Plan Finish	Displays the finish date for the task.
Plan Work	Indicates the number of hours estimated to complete this task.
Actual Start	Displays the actual start date calculated from work record entry.

Actual Finish	Displays the actual finish date task status changes to Finished .
Actual Work	Indicates the actual number of hours taken to complete the task, calculated from Work Records.
Variance Start	Displays the variance in days between expected start date and actual start date. This is read-only calculated value.
Variance Finish	Displays the variance in days between expected finish date and actual finish date. This is read-only calculated value.
Variance Work	Displays the variance in number of hours between estimated and actual duration. This is read-only calculated value.

Work

The **Work** page lists all the work records entered for this task. Each work record has the following properties.

Property	Description
User Name	Displays the name of person who performed the work for this work record entry.
Date	Displays the date of work record entry.
Work Hours	Indicates the number of hours worked for this work record entry.
Remaining Work	Indicates the remaining number of hours left to complete the task.
Comments	Displays text comments explaining what work was done for this work record.
Total Actual Work	Displays read-only calculated field of the total time spent on this task based on the work records entered.

Notes

The **Notes** page is a simply a text box for capturing notes about the task.

Custom

You can create custom properties for an item which will display in the item **Properties** dialog box.

The following properties are on the **Custom** page.

Property	Description
Property	Displays each custom property name.
Value	Displays the values for each custom property. Double-click the property name to edit the value.

Attachments

The **Attachments** page contains a list of all the files attached to the current item.

Comment

The following properties are on the **Comment** page.

Property	Description
Comment For This Revision	Displays the reason for the changes to the current revision.
Comment For New Revision	Displays the reason for the changes to the new revision.

Related Procedures

[Viewing or Modifying Item Properties](#)

Related Reference

[Change Request Properties](#)

[File Properties](#)

[Folder Properties](#)

[Requirement Properties](#)

[Topic Properties](#)

Topic Properties

This topic presents the topic properties and their descriptions as displayed in the **Topic Properties** dialog box. The **Topic Properties** dialog box contains the following tabbed pages of properties.

Topic

The following properties are on the **Topic** page.

Property	Description
Title	Displays the title of the topic.
Created By	Displays the name of person who created the topic.
Created On	Displays the date on which topic was created.
Attachments	Displays the number of attachments to the topic.
Modified By	Displays the name of the last person who modified the topic.
Modified On	Displays the date on which the topic was last modified
Content	Displays the text contents of the topic.

Options

The following properties are on the **Options** page.

Property	Description
Recipients	Displays the list of intended recipients of the topic or response. Note: You cannot delete yourself as a recipient unless you delete all the recipients. When there are recipients, StarTeam does not allow you to remove yourself from the notification list.
Priority	Displays the topic priority: Low , Normal , or High
Status	Displays the topic status: Active or Inactive .

Custom

You can create custom properties for an item which will display in the item **Properties** dialog box.

The following properties are on the **Custom** page.

Property	Description
Property	Displays each custom property name.
Value	Displays the values for each custom property. Double-click the property name to edit the value.

Attachments

The **Attachments** page contains a list of all the files attached to the current topic.

Comment

The following properties are on the **Comment** page.

Property	Description
Comment For This Revision	Displays the reason for the changes to the current revision.
Comment For New Revision	Displays the reason for the changes to the new revision.

Related Procedures

[Viewing or Modifying Item Properties](#)

Related Reference

[Change Request Properties](#)

[File Properties](#)

[Folder Properties](#)

[Requirement Properties](#)

[Task Properties](#)

Project Properties Dialog Box

Describe options in the **Project Properties** dialog box.

To access the **Project Properties** dialog box, choose **Project** ► **Properties**.

In This Section

[Name \(Project Properties Dialog Box\)](#)

[Options \(Project Properties Dialog Box\)](#)

[Default Types \(Project Properties Dialog Box\)](#)

[Process Rules \(Project Properties Dialog Box\)](#)

[Editors \(Project Properties Dialog Box\)](#)

[Change Package Properties](#)

Lists and defines the change package properties.

[Versioning Object Properties](#)

Lists and defines the common properties of all versioning properties for objects that use versioning.

Name (Project Properties Dialog Box)

[Project](#) ▶ [Properties](#) ▶ [Name](#)

Describes the options on the **Name** page of the **Project Properties** dialog box.

Item	Description
Name	Specifies the project name.
Description	Specifies the project description.
Created By	Displays the name of the person who created the project.
(Created) On	Displays the date on which the project was created.
Type	Displays the project type.

Related Procedures

[Viewing or Modifying Project Properties](#)

Related Reference

[Project Properties Dialog Box](#)

[Options \(Project Properties Dialog Box\)](#)

[Options \(Project Properties Dialog Box\)](#)

[Process Rules \(Project Properties Dialog Box\)](#)

[Editors \(Project Properties Dialog Box\)](#)

Options (Project Properties Dialog Box)

Project ► Properties ► Options

Describes the options on the **Options** page of the **Project Properties** dialog box.

Item	Description
Keyword Expansion	Enables keyword expansion.
Expand Keywords For These File Extensions	Restricts keyword expansion to files with the specified extensions.
Require Revision Comment When Files Are Checked In	Specifies that revision comments are required when checking in files to the project.
Require Exclusive Lock When Files Are Checked In	<p>Specifies that all files checked in the must be exclusively locked to enable check-in.</p> <p>If developers are using an application integration for a development environment, such as StarTeam Visual Studio Integration, they cannot check in files from that environment if both the Require Exclusive Lock When Files are Checked In check box in the Project Properties dialog box, and the Use Non-exclusive Locks in Integrations check box on the Personal Options dialog box (File tab) are selected. In this situation, uncheck Use Non-exclusive Locks in Integrations to check files in.</p>
Mark Unlocked Working Files Read-Only	<p>Specifies that all unlocked working files are marked as read-only if they are checked in, checked out, or unlocked when file locking is required. This option applies to files that are unlocked in the application or in application integrations with third-party applications.</p> <p>The project property overrides the identical Mark Unlocked Working Files Read-only personal option. See the topic "Marking Unlocked Files Read-only" in the links below.</p>

Related Procedures

[Viewing or Modifying Project Properties](#)
[Requiring Revision Comments](#)
[Enabling Keyword Expansion](#)
[Marking Unlocked Files Read-only](#)
[Requiring Exclusive Locks for Check-ins](#)

Related Reference

[Project Properties Dialog Box](#)
[Name \(Project Properties Dialog Box\)](#)
[Options \(Project Properties Dialog Box\)](#)
[Process Rules \(Project Properties Dialog Box\)](#)
[Editors \(Project Properties Dialog Box\)](#)
[Table of StarTeam Keywords](#)

Default Types (Project Properties Dialog Box)

[Project](#) ▶ [Properties](#) ▶ [Default Types](#)

Describes the options on the **Default Types** page of the **Project Properties** dialog box.

Select the item types to be included by default when creating a new view in a project. The new view will include items of the selected types from the parent view.

You can select any or none of the following item types:

- ◆ Change Request
- ◆ File
- ◆ Requirement
- ◆ Task
- ◆ Topic

Related Procedures

[Creating Projects](#)

[Viewing or Modifying Project Properties](#)

Related Reference

[Project Properties Dialog Box](#)

[Options \(Project Properties Dialog Box\)](#)

[Process Rules \(Project Properties Dialog Box\)](#)

[Editors \(Project Properties Dialog Box\)](#)

Process Rules (Project Properties Dialog Box)

Project ▸ Properties ▸ Process Rules

Describes the options on the **Process Rules** page of the **Project Properties** dialog box.

Item	Description
Require Selection Of Process Items When Files Are Added Or Checked In	Enforces selection of process items when files are added or checked in.
Permit Selection Of Change Requests As Process Items	<p>Permits the selection of change requests as process items, and restricts them based on which status fields are checked in the Restrict Status To check boxes:</p> <p>If Open is checked, only change requests with a status of Open can be used as process items.</p> <p>If In Progress is checked, only change requests with a status of In Progress can be used as process items.</p> <p>If both Open and In Progress are checked, only change requests with a status of Open or In Progress can be used as process items.</p> <p>If neither Open and In Progress are checked, all change requests can be used as process items, regardless of their status.</p>
Permit Selection Of Requirements As Process Items	<p>Permits the selection of requirements as process items, and restricts them based on which status fields are checked in the Restrict Status To check boxes:</p> <p>If Approved is checked, only requirements with a status of Open can be used as process items.</p> <p>If Approved is unchecked, all requirements can be used as process items, regardless of their status.</p>
Permit Selection Of Tasks As Process Items	<p>Permits the selection of tasks as process items, and restricts them based on which status fields are checked in the Restrict Status To check boxes:</p> <p>If Ready To Start is checked, only tasks with a status of Ready To Start can be used as process items.</p> <p>If In Progress is checked, only tasks with a status of In Progress can be used as process items.</p> <p>If both Ready To Start and In Progress are checked, only tasks with a status of Ready To Start or In Progress can be used as process items.</p> <p>If neither Ready To Start and In Progress are checked, all tasks can be used as process items, regardless of their status.</p>
Enable Enhanced Process Links	<p>Enables the client to use enhanced process links.</p> <p>If checked, the process item (that is, the item specified as the reason for making a given set of changes) is distinguished from the task that represents the act of making the associated changes in a particular view. Changes are linked to the process item <i>indirectly</i>, through a process task which is automatically created by the client.</p>

If unchecked, standard links are used where the source of a process link is itself a process item; that is, if a given item is specified as the reason for a change, then process links are created *directly* from that process item to each changed file or folder.

Related Concepts

[Process Rules](#)

[Process Items and Process Links](#)

[Process Tasks and Enhanced Process Links](#)

Related Procedures

[Viewing or Modifying Project Properties](#)

Related Reference

[Project Properties Dialog Box](#)

[Name \(Project Properties Dialog Box\)](#)

[Options \(Project Properties Dialog Box\)](#)

[Options \(Project Properties Dialog Box\)](#)

[Editors \(Project Properties Dialog Box\)](#)

Editors (Project Properties Dialog Box)

Project ► Properties ► Editors

The check box options on the **Editors** page of the **Project Properties** dialog box enable you to specify alternate property editors (APEs) for editing the item properties.

Item	Description
Use Alternate Property Editor For Files	Specifies that for files, StarTeam should use the specified alternate property editor in the corresponding field.
Use Alternate Property Editor For Change Requests	Specifies that for change requests, StarTeam should use the specified alternate property editor in the corresponding field.
Use Alternate Property Editor For Requirements	Specifies that for requirements, StarTeam should use the specified alternate property editor in the corresponding field.
Use Alternate Property Editor For Tasks	Specifies that for tasks, StarTeam should use the specified alternate property editor in the corresponding field.
Use Alternate Property Editor For Topics	Specifies that for topics, StarTeam should use the specified alternate property editor in the corresponding field.

Note: Your company must use StarTeam Enterprise Advantage to be able to use APEs. For more information creating APEs, see the "StarTeam Extensions User's Guide" in the StarTeam documentation.

Related Procedures

[Viewing or Modifying Project Properties](#)

[Configuring Projects to use APEs](#)

Related Reference

[Project Properties Dialog Box](#)

[Name \(Project Properties Dialog Box\)](#)

[Options \(Project Properties Dialog Box\)](#)

[Options \(Project Properties Dialog Box\)](#)

[Process Rules \(Project Properties Dialog Box\)](#)

Change Package Properties

This topic presents the change package properties and their descriptions as displayed in the **Change Package Properties** dialog box. The **Change Package Properties** dialog box contains the following tabbed pages of properties.

Change Package

The following properties are available for **Change Packages**. User's with the proper access rights can change these properties.

Property	Description
Name	Displays the textual identification of the change package. The name must be unique among all other change packages for the same target view. By default, StarTeam generates a name that contains a timestamp, which helps prevent duplicate names. This property is the change package object's <i>primary descriptor</i> .
Description	Provides a full description of the change package. A typical use of the description property is to provide documentation not available in the change package's process item such as notes to testers, reviewers, or developers.
Session Type	Defines the change package's basic type. The type is set when the change package is first created and cannot be changed thereafter. It has the same values as VCM: Rebase, Replicate, and Promote.
Target View	Displays the name of the change package's target view.
Source View	Displays the name of the change package's source view.
Commit Time	Indicates the timestamp at which the change package was committed, or <code>null</code> if it has not yet been committed. This property cannot be directly modified by the user. Change packages that have been committed cannot be modified (except for the revision comment).
State	<p>Indicates the current workflow state of the change package. For the Hamachi release, this property will not be directly user-modifiable nor user-customizable. Furthermore, change packages will not be integrated with user-customizable workflow, hence the values of this property are defined by actions performed to the change package as it evolves. The initial enumeration values of the state property are:</p> <p>New: This is the state that every change package receives when it is first created, but before any updates are defined for it.</p> <p>Committed:: This state indicates that the change package has been committed, hence its changes have been applied. A committed change package and can no longer be modified (except for the revision comment).</p>
Transaction ID	Holds the ID of the transaction in which the change package's updates were applied. Prior to the <i>Committed</i> state, its value is <code>null</code> . Otherwise, the value can be used to query audit records in the Audit tab or commands used in the transaction by way of the StarTeam server command trace file.
Pre-commit View Label	Displays the name of the pre-modification revision label that was created in the target view. This property is <code>null</code> if this label type was not created.
Pre-commit Revision Label	Displays the name of the pre-modification revision label that was created in the target view. This property is <code>null</code> if this label type was not created.
Post-commit View Label	Displays the name of the post-modification view label that was created in the target view. This property is <code>null</code> if this label type was not created.
Post-commit Revision Label	Displays the name of the post-modification revision label that was created in the target view. This property is <code>null</code> if this label type was not created.

Committed by	Displays the name of the user who committed the change package or null if it has not yet been committed.
Committed In Build	<p>Displays the name of the first build label created in the same view after the change package is committed of null if it has not yet been committed.</p> <p>When a change package is committed, this property displays Next Build until a new build label is created.</p>
Responsibility	<p>Displays the name of the user who is currently responsible for the change package. It is initially set to the user that first creates the change package.</p> <p>If responsibility is changed to another user, and email notifications are enabled, the new user is notified of the change.</p>

Other Server Services

In addition to the StarTeam server's persistence and versioning service, change packages also use the server's locking service. This means that a change package can be locked exclusively or with a shared lock. An exclusive lock is automatically applied to a saved change session when it is opened for editing. A change session can also be opened in read-only mode, though a non-exclusive lock is not applied for this use.

Note: Change packages cannot be "flagged" (bookmarked).

Related Reference

[Change Package Properties](#)

Versioning Object Properties

Lists and defines the common properties of all objects that use versioning. For reference, these common properties are summarized below:

External (Internal) Property Name	Type	Description
Comment (Comment)	text	This calculated property provides the object's revision comment regardless of whether it is stored in the ShortComment or LCOMM table.
CommentID (CommentID)	int	If a revision comment < 2,000 characters exists, this field is a foreign key to the LCOMM table that contains the full comment.
Created By (CreatedUserID)	int	ID of the user that first created the object.
Created Time (CreatedTime)	timestamp	Date/time at which the object was created.
Deleted By (DeletedUserID)	int	ID of the user that deleted the object. -1 means the object is still "live".
Deleted Time (DeletedTime)	timestamp	Date/time that the object was deleted. 0 means object is still "live". Non-zero means the record is soft-deleted.
Dot Notation (DotNotation)	text	This calculated property is the dotted-notation representation of the object's version.
End Modified Time (EndModifiedTime)	timestamp	Date/time that this object revision was replaced by another revision. 0 identifies the latest (tip) revision.
Locked By (ExclusiveLocker)	int	ID of the user that currently has the object exclusively locked. -1 means the object is not exclusively locked.
Modified By (ModifiedUserID)	int	ID of the user that last modified the object.
Modified Time (ModifiedTime)	timestamp	Date/time that the object was last modified.
My Lock (MyLock)	int	This client-calculated property indicates if the current user has this object locked.
New Revision Comment (NewRevisionComment)	text	This client-calculated property serves as a place holder for a new revision comment value.
Non-Exclusive Lockers (NonExclusiveLockers)	text	List of users that currently have the object non-exclusively locked, if any.
Object ID (ID)	int	ID that is unique to each object in the database.
Read Only (ReadOnly)	int	This calculated property indicates if the object is currently read-only.
Revision Flags	int	<i>This field is currently always zero.</i>

(RevisionFlags)		
Short Comment	text	Value of the revision comment if it is less than 2,000 characters.
(ShortComment)		
Version	int	Revision number of this object revision. 0 = first revision.
(RevisionNumber)		

Related Reference

[Change Package Properties](#)

Change Requests

This section contains reference topics related to change requests.

In This Section

[Default and Required Change Request Fields](#)

Table listing the available change request fields, explains their use, and indicates which are required.

[Commonly Used Change Request Abbreviations](#)

Describes common syntax used when entering change requests.

Default and Required Change Request Fields

The following table lists the fields on the **Change Request** dialog, explains their uses, and indicates which fields are required. In the table, CR is used as an acronym for change requests.

Field	Required?	Description	Example
Status	Y	For new CRs, set the Status field to New . The Status is changed to Open when the CR is assigned to a developer.	In this example, the status should be New .
Severity	Y	Specify the seriousness of the problem. High severity items are usually associated with data loss or corruption, system crashes, etc. Low severity items are generally misspelled items and cosmetic errors.	In this example, the problem is comparatively minor (that is, if it does not cause the system to crash or lose data), so classify it as Medium .
Priority	Y	In most defect tracking systems, Priority is a multi-level choice (usually on a 1 to 5 scale). In StarTeam, however, it is a Yes or No choice. The priority of a CR is sometimes determined by the tester and sometimes by the developer. In most cases, it reflects the need to get a particular defect fixed before others. If the defect is catastrophic or prevents your team from accessing other major areas of the application, select the Priority field.	In this example, leave the Priority field cleared.
Platform	Y	Indicate what type of operating system environment the defect occurs in. If the defect happens only on Windows NT, select Windows NT. In most cases, the defect will appear on all platforms.	In this example, set Platform to All .
External Reference	N	Specify information received from outside the company, such as a note about a defect from an outside testing service or a customer. Currently this field is not used.	In this example, leave the field empty.
Component	N	Identify the component of the product in which the defect occurs. Currently this field is not used.	In this example, leave the field empty.
Category	N	Identify a subcomponent of the product. It is used with the Component field to identify the location in which the defect occurs. Currently this field is not used.	In this example, leave the field empty.
Synopsis	Y	Use to give a brief summary of the problem encountered or the suggested enhancement. Consider the synopsis to be a title for the defect. Note: The Synopsis should only contain information for one defect. If	For this example, a synopsis might be: "Available fields disappear when using the Advanced Fields box."

		the reported defect uncovers or relates to another defect, the second defect should be written up separately and referenced to the first defect in the synopsis (for example, " CR #3109 also relates to this defect ").	
Type	Y	If the CR is a reproducible problem in the software, select Defect . If it is a customer request or a feature enhancement request, select Suggestion .	For this example, select Defect .
Last Build Tested	Y	Indicate the build number of the software in which the defect was discovered or last tested. If you are writing a CR, select the build number from the application (often found in the About dialog). If you are verifying or regressing the CR, and the problem still exists in the current build, change this field to the build number you are currently testing.	For this example, select the most current build number.
Addressed in Build	Y	Indicate the build in which the fix first appears. In most cases, after the engineer fixes the defect, the field will be set to Next Build . This field changes to the correct build when that version is actually built.	For this example, leave the field empty.
Responsibility	N	Indicate the person who should act on the defect. Depending on the position of the CR in the CR life cycle, this person could be a developer, a QA engineer, or the person who first reported the CR.	For this example, either leave this field blank, or assign it to the lead engineer on the project, who will assign it to the appropriate person.
Addressed by	Y	This field is automatically filled with the name of the person who originally wrote up the CR. It is not editable.	NA
Description/Steps to Reproduce	Y	<p>Select the Description Tab. In the Description/ Steps to Reproduce field, enter detailed information about the defect. Specifically, the description should build on the synopsis information.</p> <p>The Steps to Reproduce information is the most important data entered in the CR because it provides a detailed step-by-step method of reproducing the defect. The more detailed the information, the more likely the responsible developer will be able to determine the cause of the defect and fix the defect.</p>	<p>Steps to Reproduce might look as follows:</p> <ol style="list-style-type: none"> 1. RClick the column headers in the upper pane. 2. Select Show Fields. 3. LClick Show Advanced Fields check box. // The check box is activated. 4. LClick Show Advanced Fields check box. // The check box is deactivated. 5. EXP: The standard fields appear in the Available Fields list box. 6. ACT: No fields appear in the Available Fields list box.

Related Concepts

[Change Requests](#)

Related Procedures

[Working with Change Requests](#)

Related Reference

[Commonly Used Change Request Abbreviations](#)

Commonly Used Change Request Abbreviations

When using a defect tracking system, most organizations employ a special syntax or shorthand to describe the steps required to reproduce a problem. The following table lists a recommended shorthand and describes each item in that shorthand.

Shorthand	Description
1, 2, 3,...	Number the steps.
LClick	Click the left mouse button.
RClick	Click the right mouse button.
DClick	Double click the left mouse button.
[]	A keyboard button to be pressed. For example, [F1] for Help or [F5] for Refresh.
< >	A dialog button. For example: Press <OK> or <Cancel>.
>	Menu separator. For example: Select File > Open or Topic > Tools > Reports.
//	A comment that is not an actual step. For example: 3) LClick the field. // At this point my machine started to smoke.
EXP	Expected results. For example: EXP: The focus moves to the next field.
ACT	Actual results. For example: ACT: The application crashes.

Related Concepts

[Change Requests](#)

Related Procedures

[Working with Change Requests](#)

Related Reference

[Default and Required Change Request Fields](#)

Charts

This section contains reference topics related to charts.

In This Section

[Chart Types](#)

Describes the chart types available in StarTeam.

[Default Chart Views and Zoom/Rotate Capabilities](#)

Table of default chart views and their zooming/rotating capabilities.

[File Chart Fields](#)

Table listing the available fields for the different types of file charts.

[Change Request Chart Fields](#)

Table listing the available fields for the different types of change request charts.

[Requirement Chart Fields](#)

Table listing the available fields for the different types of requirement charts.

[Task Chart Fields](#)

Table listing the available fields for the different types of task charts.

[Topic Chart Fields](#)

Table listing the available fields for the different types of topic charts.

[Audit Chart Fields](#)

Table listing the available fields for the different types of audit charts.

Chart Types

The table below lists the available chart types and their descriptions.

Chart Type	Description
Area	<p>Area charts emphasize the amount of change over a period of time, or they compare multiple items. An area chart also shows the relationship of parts to a whole by displaying the total of the plotted values.</p> <p>An area chart is a form of line chart, but the area between the horizontal (x) axis and the line connecting the data markers is filled with color. This makes it easy to see where the points encompassed by the different data series overlap.</p>
Bar	<p>Bar Charts show the changes in a data series over time, or they compare multiple items. Types of items are arranged vertically and data values are plotted horizontally to emphasize variation over time.</p> <p>The 3-D bar chart provides an extra dimension for plotting data by comparing values along two axes.</p>
Bubble	<p>Bubble charts are a type of scatter chart. The x and y coordinates of the data marker (the bubble) are determined by two data values. The size of the data marker indicates the value of a third variable.</p>
Column	<p>Column charts show the changes in a data series over time, or they compare multiple items. Types of items are arranged horizontally and data values are plotted vertically to emphasize variation over time.</p> <p>The 3-D column chart provides an extra dimension for plotting data by comparing values along two axes.</p>
Heat Map	<p>Heat Map charts show the relationship between data items by using gradually changing shades of color. Heat map charts are commonly used in financial analysis to show which stocks are rising, which are falling and the amount and rate of change between them.</p>
Line	<p>Line charts emphasize the amount of change over a period of time, or they compare multiple items. Data points are plotted in series using evenly-spaced intervals and connected with a line to emphasize the relationships between the points.</p>
Pie	<p>Pie charts show the size of items that make up a data series proportional to the total of the items in the series. Pie charts always show a single data series, and it is useful for determining which items in the series are most significant.</p>
Scatter	<p>Scatter charts are used either to show the relationship of items in several distinct series of data, or to plot two sets of values as one series of x/y coordinates. Scatter chart draw attention to uneven intervals or clusters of data. This type of chart is often used to plot scientific data, and can highlight the deviation of collected data from predicted results.</p>
Stack Bar	<p>Stack Bar charts show the relationship of individual items in a series to the whole.</p>
Stack Column	<p>Stack Column charts show the relationship of individual items in a series to the whole.</p>

Related Procedures

[Generating Simple Charts](#)
[Generating Distribution Charts](#)
[Generating Correlation Charts](#)
[Generating Time-series Charts](#)
[Viewing Charts](#)
[Choosing the Chart Type](#)
[Exporting a Chart as an Image](#)
[Configuring Chart Colors](#)
[Customizing Chart Titles](#)

Related Reference

[Audit Chart Fields](#)
[Change Request Chart Fields](#)
[File Chart Fields](#)
[Requirement Chart Fields](#)
[Task Chart Fields](#)
[Topic Chart Fields](#)

Default Chart Views and Zoom/Rotate Capabilities

Not all charts are capable of a 3D display, nor can you zoom in and out of, or rotate, all types of charts. For example, you cannot zoom or rotate any chart while it is in a 2D view.

The table below displays all the chart types and lists their default and available views, and their zooming and rotating capabilities.

Chart	3D - Zoom, Rotate	2D - Zoom, Rotate	Default View
Area	Yes	No	3D
Bar	Yes	No	3D
Column	Yes	No	3D
Bubble	No	No	2D (3D not available)
Heat Map	Yes	No	3D
Line	Yes	No	3D
Pie	Yes	No	3D
Scatter	No	No	2D (3D not available)
Stack Bar	Yes	No	3D
Stack Column	Yes	No	3D

Related Procedures

[Generating Simple Charts](#)

[Generating Distribution Charts](#)

[Generating Correlation Charts](#)

[Generating Time-series Charts](#)

[Viewing Charts](#)

File Chart Fields

This table lists the available fields for file charts.

Simple	Distribution	Correlation	Time-series
Content Revision	Create charts based on fields currently being grouped.	Content Revision	Configuration Time
DotNotation ID	—	DotNotation ID	Created Time
EOL Character	—	EOL Character	Deleted Time
Object ID	—	Object ID	File Time Stamp at Check-in
Parent ID	—	Parent ID	Modified Time
Parent Branch Revision	—	Parent Branch Revision	Sync Local Time Stamp
Parent Revision	—	Parent Revision	Working File Time Stamp
Project ID	—	Project ID	—
Revision	—	Revision	—
Revision on Disk	—	Revision on Disk	—
Root Object ID	—	Root Object ID	—
Size	—	Size	—
Sync Branch Version	—	Sync Branch Version	—
Sync Content Version	—	Sync Content Version	—
Sync Local Size	—	Sync Local Size	—
Sync on Path to Root	—	Sync on Path to Root	—
Vault Branch Version	—	Vault Branch Version	—
Version	—	Version	—
Working File Size	—	Working File Size	—

Related Reference

[Chart Types](#)

[Audit Chart Fields](#)

[Change Request Chart Fields](#)

[Requirement Chart Fields](#)

[Task Chart Fields](#)

[Topic Chart Fields](#)

Change Request Chart Fields

This table lists the available fields for change request charts.

Simple	Distribution	Correlation	Time-series
Attachment Count	Create charts based on fields currently being grouped.	Attachment Count	Closed On
CR Number	—	CR Number	Configuration Time
DotNotation ID	—	DotNotation ID	Created Time
Object ID	—	Object ID	Deleted Time
Parent ID	—	Parent ID	Entered On
Parent Branch Revision	—	Parent Branch Revision	Modified Time
Parent Revision	—	Parent Revision	Resolved On
Root Object ID	—	Root Object ID	Total Open
Version	—	Version	Verified On

Related Reference

[Chart Types](#)

[Audit Chart Fields](#)

[File Chart Fields](#)

[Requirement Chart Fields](#)

[Task Chart Fields](#)

[Topic Chart Fields](#)

Requirement Chart Fields

This table lists the available fields for requirement charts.

Simple	Distribution	Correlation	Time-series
Ambiguities Found	Create charts based on fields currently being grouped.	Ambiguities Found	Created Time
Attachment Count	—	Attachment Count	Configuration Time
Children Count	—	Children Count	Deleted Time
Comment ID	—	Comment ID	Modified Time
Expected Effort	—	Expected Effort	End Modified Time
High Effort	—	High Effort	—
Low Effort	—	Low Effort	—
Number	—	Number	—
Object ID	—	Object ID	—
Parent	—	Parent	—
Requirement ID	—	Requirement ID	—
Responsible Count	—	Responsible Count	—
Revision Flags	—	Revision Flags	—
Version	—	Version	—

Related Reference

[Chart Types](#)

[Audit Chart Fields](#)

[Change Request Chart Fields](#)

[File Chart Fields](#)

[Task Chart Fields](#)

[Topic Chart Fields](#)

Task Chart Fields

This table lists the available fields for task charts.

Simple	Distribution	Correlation	Time-series
Actual Hours	Create charts based on fields currently being grouped.	Actual Hours	Actual Finish
Attachment Count	—	Attachment Count	Actual Start
Children Count	—	Children Count	Configuration Time
Estimated Hours	—	Estimated Hours	Constraint Date
MS Task Unique ID	—	MS Task Unique ID	Created Time
Object ID	—	Object ID	Deleted Time
Parent Task ID	—	Parent Task ID	Estimated Finish
Percent Complete	—	Percent Complete	Estimated Start
Task Duration	—	Task Duration	Last MS Project Update
Task Number	—	Task Number	Last Work/Dependency Update
Version	—	Version	Modified time

Related Reference

[Chart Types](#)
[Audit Chart Fields](#)
[Change Request Chart Fields](#)
[File Chart Fields](#)
[Requirement Chart Fields](#)
[Topic Chart Fields](#)

Topic Chart Fields

This table lists the available fields for topic charts.

Simple	Distribution	Correlation	Time-series
Attachment Count	Create charts based on fields currently being grouped.	Attachment Count	Created Time
Children Count	—	Children Count	Configuration Time
Recipient Count	—	Recipient Count	Deleted Time
Object ID	—	Object ID	Modified Time
Parent Topic ID	—	Parent Topic ID	—
Topic Number	—	Topic Number	—
Version	—	Version	—

Related Reference

[Chart Types](#)

[Audit Chart Fields](#)

[Change Request Chart Fields](#)

[File Chart Fields](#)

[Requirement Chart Fields](#)

[Task Chart Fields](#)

Audit Chart Fields

This table lists the available fields for audit charts.

Simple	Distribution	Correlation	Time-series
Not Available	Create charts based on fields currently being grouped.	Not Available	Created Time
—	—	—	Deleted Time
—	—	—	Modified Time

Related Reference

[Chart Types](#)

[Change Request Chart Fields](#)

[File Chart Fields](#)

[Requirement Chart Fields](#)

[Task Chart Fields](#)

[Topic Chart Fields](#)

Exporting

Describes the options for exporting items.

In This Section

[Export Dialog Box Options](#)

This topic describes the options for exporting data as delimited text for use in other applications.

Export Dialog Box Options

To specify export options, open the **Export** dialog box. Right-click in the upper pane and choose [Advanced Export](#).

Use the **Export** dialog box to export data in the upper pane to a text-delimited file for import into other applications, such as spreadsheets and database applications. Below are the fields and options in the **Export** dialog box.

Item	Description
Available Fields	Lists all the fields displayed as columns in the upper pane which are available for export. You are limited to displaying 60 columns in the upper pane, and therefore, are limited to exporting 60 columns of data. The target application must be able to import the number of fields you are exporting.
Advanced Fields	Displays the advanced fields in the Available Fields list.
Show These Fields In This Order	Lists the fields you selected for export. The fields will be exported in the order they appear in this list.
Current Selection	Specifies that the export should only include the items currently selected in the upper pane on the active component tab .
All Items	Specifies that the export should include all the items in the upper pane on the active component tab.
Export Field Names	Includes the column header fields for the report.
Replace Embedded Carriage Returns With Spaces	Exports the data without embedded carriage returns/line feeds.
Text Qualifier	Specifies the text qualifier to use at the beginning and end of text fields for preventing the special treatment of carriage returns/line feeds and other special characters that occur in HTML output.
Delimiter	Specifies the character to use to separate the data fields in the exported text file. You may enter a custom delimiter or choose from the predefined list.
Encoding	Specifies the encoding to use for the exported data.
Output File Name	Specifies the location and name of the output text file. The file must have a <code>.txt</code> extension, or another extension the target application will import. The default extension is <code>.txt</code> .

Related Concepts

[Fields](#)

Related Procedures

[Exporting Data](#)

[Displaying Additional Fields](#)

Related Reference

[Fields](#)

Fields

Detailed lists of the fields found in the upper pane on each of the component tabs, and relational operators used in queries.

In This Section

[File Fields](#)

[Change Request Fields](#)

[Requirement Fields](#)

[Task Fields](#)

[Topic Fields](#)

[Folder Fields](#)

[Audit Fields](#)

[Relational Operators Used in Queries](#)

Describes the relational operators that you can use to define conditions in a query.

File Fields

This section lists all the file fields in alphabetical order.

Note: Client-calculated fields cannot be used in custom email notifications or Notification Agent. Reports can use any field name.

Field	Description
Archive Format (Advanced)	Values: Native-II Internal Identifier: ArchiveFormat Indicates whether the format in which file revisions are stored. StarTeam 2006 uses only Native-II format.
Archive Path	Values: text Internal Identifier: ArchivePath The path to the PVCS archive or VSS project containing a file.
Archive Name (Advanced)	Values: 32-digit hex string representing the MD5 value of the file Internal Identifier: STArchiveName For a file stored in a Native-II Vault, indicates the name of the file that stores the tip revision. This name is the MD5 value of that file revision's content, converted to a 32-digit hex string.
Archive Type	Values: Native, PVCS , VSS Internal Identifier: Type Indicates whether a file is stored as a StarTeam (Native), PVCS, or VSS file.
Branch On Change (Advanced)	Values: No Yes Internal Identifier: BranchOnChange Indicates whether a file will branch when it changes. The value is No if the file's behavior is not set to Branch On Change . Reasons for this may be: <ul style="list-style-type: none">- The file is in the root or a reference view and the Branch On Change feature is disabled.- The file is in a branching view but has already branched as a result of a change, which, in turn, results in the Branch On Change feature becoming disabled.- The file is in a branching view, but its behavior currently does not permit it to branch on change. This means that modifications are checked into the parent view. Note: If the value is No , the value of the Branch State explains the No .
Branch State (Advanced)	Values: Branched , Not Branched , Root Internal Identifier: BranchState Indicates whether a file has branched in the child view, is still unbranched (and therefore is part of the parent view), or was created in the view in which it resides.

	<p>The values Branched and Not Branched apply to files in branching views. The value Root applies to files created in the view in which the file currently resides.</p> <p>If the view is a reference view, it reflects the state of the file in the reference view's parent.</p>
Comment	<p>Values: text</p> <p>Internal Identifier: Comment</p> <p>The initial 2000 characters provided as the reason for changing a file's properties or contents are stored in the Short Comment field. The Comment field stores those 2000 characters and any additional text. Changing a file's properties causes the application to create a new revision. Note: To include a Link comment, the Comment field is the value to use in an HTML Report.</p>
CommentID (Advanced)	<p>Values: number</p> <p>Internal Identifier: CommentID</p> <p>The ID number assigned to revision comment. Displays -1 if no revision comment was supplied.</p>
Compression Level	<p>Values: Default, Maximize Compression, Maximize Speed, None</p> <p>Internal Identifier: Compression</p> <p>Indicates a file's level of compression.</p> <p>Default: A compromise between Maximize Compression and Maximize Speed.</p> <p>Maximize Compression: The densest possible compression of file revisions to improve server performance.</p> <p>Maximize Speed: The fastest possible compression of file revisions to improve server performance.</p> <p>None: No compression.</p>
Configuration Time	<p>Values: date/time</p> <p>Internal Identifier: ConfigurationTime</p> <p>Indicates the time to which a file is configured. If you configure a file to a specific time, this field contains that time. If you configure a file to a label or promotion state, this field shows either the time at which the label was created, or the time at which the label associated with the promotion state was created.</p>
Content Revision	<p>Values: number</p> <p>Internal Identifier: ContentVersion</p> <p>The number of times a file has been checked in. If the file is in a child view, it includes all the content revisions from the parent view in this number. Each revision appears in the file's history.</p>
Created By	<p>Values: list of users, <None></p> <p>Internal Identifier: CreatedUserID</p> <p>The name of the user who created the first revision in the view. This is either the user who added the file to the project, or the user who checked in the revision that branched.</p>
Created Time	<p>Values: date/time</p>

	Internal Identifier: CreatedTime
	The time at which the first revision in the view was created.
Deleted By	Values: list of users, <None>
	Internal Identifier: DeletedUserID
	The name of the user who deleted a file. Because deleted files do not appear in the list, this information is unavailable to users. Internal Use Only.
Deleted Time	Values: date/time
	Internal Identifier: DeletedTime
	The time at which a file was deleted. Because deleted files do not appear in the list, this information is unavailable to users. Internal Use Only.
Description	Values: text
	Internal Identifier: Description
	The description provided for a file at the time it was added to the view, including any later edits to it.
Dot Notation	Values: text
	Internal Identifier: DotNotation
	The branch revision number, for example, 1.2.1.0 .
End Modified Time (Advanced)	Values: date/time
	Internal Identifier: EndModifiedTime
	The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.
EOL Character	Values: numeric representation of ANSI character
	Internal Identifier: EOL
	For internal use only. This field is primarily used to determine an ANSI character to use when breaking up lines within files for delta storage with Native-1 Vaults.
Executable	Values: No , Yes
	Internal Identifier: Executable
	Indicates whether the executable bit should be set for a UNIX file.
Extension	Values: text
	Internal Identifier: Extension
	Displays the extension of the file. This field is client-calculated.
File Time Stamp at Check-In	Values: date/time
	Internal Identifier: Modified
	The file's time stamp at the time it was last checked in.
File Type	Values: ASCII , Binary , Unicode
	Internal Identifier: Charset
	Indicates whether the file is an ASCII (text), binary, or Unicode.
Flag	Values: No , Yes
	Internal Identifier: Flag

	Marks or bookmarks files in the upper pane on your workstation. This is a client-calculated field.
Flag User List (Advanced)	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.</p> <p>Internal Identifier: <code>FlagUserList</code></p> <p>Cannot be used in queries. Identifies users who have set flags on a given item.</p>
Folder	<p>Values: text</p> <p>Internal Identifier: <code>Folder</code></p> <p>The name of the folder with which a file is associated. This is not the name of the working folder.</p>
Folder Path	<p>Values: text</p> <p>Internal Identifier: <code>Folder Path</code> (contains spaces)</p> <p>The path to the folder with which a file is associated. This is not the path to the working folder. This is a client-calculated field.</p>
Hive ID (Advanced)	<p>Values: number assigned by the Server</p> <p>Internal Identifier: <code>HiveID</code></p> <p>Indicates the ID number of the hive that stores the tip revision for a file stored in a Native-II Vault.</p>
Item Deleted By	<p>Values: list of users, <code>None</code></p> <p>Internal Identifier: <code>ItemDeletedUserID</code></p> <p>The name of the user who deleted this item. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Item Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: <code>ItemDeltedTime</code></p> <p>The time at which the item was created. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Local Name	<p>Values: text</p> <p>Internal Identifier: <code>LocalName</code></p> <p>Name of the working file. This is a client-calculated field.</p>
Local Path	<p>Values: text</p> <p>Internal Identifier: <code>LocalPath</code></p> <p>Path name to the folder containing the working file. This is a client-calculated field.</p>
Locked By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>ExclusiveLocker</code></p> <p>The name of the user who has exclusively locked a file.</p>
MD5 Checksum	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. The StarTeam client displays only significant zeroes so the <code>05</code> and <code>0A</code> would become just <code>5</code> and <code>A</code>, and <code>A-F</code> as <code>a-f</code>.</p> <p>Internal Identifier: <code>MD5</code></p> <p>Cannot be used in queries. The MD5 checksum for the tip revision.</p>
Modified By	<p>Values: list of users, <code><None></code></p>

	Internal Identifier: <code>ModifiedUserID</code>
	The name of the user who last modified a file.
Modified Time	Values: date/time
	Internal Identifier: <code>ModifiedTime</code>
	The time at which a file was last modified. The file may have been checked in or had its properties changed. This has nothing to do with the working file. Use Local Time Stamp for the time a working file was last modified.
My Lock	Values: <code>Exclusively Locked By Me</code> , <code>Non-exclusively Locked By Me</code> , <code>Not Locked By Me</code>
	Internal Identifier: <code>MyLock</code>
	Indicates whether the current user has the file locked and, if so, whether that lock is exclusive or not. This is a client-calculated field.
Name	Values: <code>text</code>
	Internal Identifier: <code>Name</code>
	Displays the name of the file.
New Revision Comment (Advanced)	Values: <code>text</code>
	Internal Identifier: <code>NewRevisionComment</code>
	Internal use only. the client uses this value during the item update process. The field always appears empty if added to the upper pane. This is a client-calculated field.
Non-Exclusive Lockers	Values: <code>text</code>
	Internal Identifier: <code>NonExclusiveLockers</code>
	The names of the users who have locked the file non-exclusively.
Object ID	Values: number
	Internal Identifier: <code>ID</code>
	Each file is assigned an object ID when it is added to a view. When it is branched in a child view, it is assigned another object ID. The original ID belongs to the file in the parent view.
Parent Branch Revision (Advanced)	Values: number
	Internal Identifier: <code>PathRevision</code>
	The last digit in the branch revision number before a file branched. For example, if this number is <code>7</code> , the branch revision was <code>1.7</code> at the time the file branched (becoming <code>1.7.1.0</code> , as seen in the file's history). This number is <code>-1</code> if a file was not inherited from the parent view.
Parent ID (Advanced)	Values: number
	Internal Identifier: <code>ParentObjectID</code>
	The object ID of a file in the parent view. The Parent ID is <code>-1</code> if this view has no parent view.
Parent Revision (Advanced)	Values:
	Internal Identifier: <code>ParentRevision</code>
	The revision number at which a file branched. For example, if this number is <code>8</code> , this file's revision number in the parent view was <code>8</code> at the time the file branched. The history should show that revision <code>9</code> in the first revision in the

	current view. This number is 0 if this file was not inherited from the parent view.
Path	<p>Values: text</p> <p>Internal Identifier: <code>Path</code></p> <p>The path to a file's working folder. This is a client-calculated field.</p>
Project ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: <code>ProjectID</code></p> <p>The ID number assigned to a project. Within a server configuration, projects are assigned ID numbers in the order in which they are created. The first project has ID 0.</p>
PVCS Revision (Advance)	<p>Values: text</p> <p>Internal Identifier: <code>PVCSRev</code></p> <p>The file's revision number in PVCS's dot notation.</p>
Read Only (Advanced)	<p>Values: <code>No</code>, <code>Yes</code></p> <p>Internal Identifier: <code>ReadOnly</code></p> <p>Indicates whether the file's configuration is read-only (as in a rollback configuration of a view) and/or its behavior does not allow it to branch on modification. Do not confuse a read-only configuration (an application issue) with a read-only file (an operating system issue). A read-only file cannot be edited and saved to disk. A file whose configuration is read-only can be edited and saved to disk; it just cannot be checked in.</p>
Revision	<p>Values: number</p> <p>Internal Identifier: <code>ViewVersion</code></p> <p>The number of times a file has been checked in or had its properties changed. If the file is in a child view, it includes all the revisions from the parent view in this number. This is a client-calculated field.</p>
Revision Flags (Advanced)	<p>Values: 0</p> <p>Internal Identifier: <code>RevisionFlags</code></p> <p>Internal use only.</p>
Revision on Disk	<p>Values: number</p> <p>Internal Identifier: <code>SyncPathVersion</code></p> <p>The number of the revision that is currently in the working folder on your workstation. The application displays no number if the file's status is Missing. This is a client-calculated field.</p>
Root Object ID (Advanced)	<p>Values: <code>number</code></p> <p>Internal Identifier: <code>RootObjectID</code></p> <p>The object ID of the oldest ancestor of a file. For example, if a file was not inherited from a parent view, the root object ID is the same as its object ID. If it was inherited from a parent view, the root object ID is the Parent ID, or the parent's Parent ID.</p>
Share State	<p>Values: <code>DerivedShare</code>, <code>Not Shared</code>, <code>Root Share</code></p> <p>Internal Identifier: <code>ShareState</code></p> <p>Indicates whether this item is shared. <code>Not Shared</code> means that the item is not shared. <code>Root Share</code> means that the item is shared and this item is the</p>

	original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.
Short Comment	<p>Values: text</p> <p>Internal Identifier: ShortComment</p> <p>Stores the initial 2000 characters provided as the reason for changing a file's properties or contents. Additional text is stored in the Comment field.</p>
Size	<p>Values: number</p> <p>Internal Identifier: FileSize</p> <p>The tip revision's size in bytes.</p>
Status	<p>Values: Current, Merge, Missing, Modified, Not In View, Out Of Date, Unknown</p> <p>Internal Identifier: Status</p> <p>Indicates the relationship between the copy of a file in your working folder and the tip revision in the repository.</p>
Storage Type (Advanced)	Obsolete
Sync Branch Version	<p>Values: number</p> <p>Internal Identifier: SyncObjectVersion</p> <p>A field used to determine status. The last number of the branch revision that was most recently checked out to the working folder. This is a client-calculated field.</p>
Sync Content Version	<p>Values: number</p> <p>Internal Identifier: SyncContentVersion</p> <p>A field used to determine status. The revision checked out as the working file or, if the file needs to be merged, a number higher than that. This is a client-calculated field.</p>
Sync Known	<p>Values: No, Yes</p> <p>Internal Identifier: SyncKnown</p> <p>A field used to determine status. Indicates whether the server knows the working file's relationship to the tip revision. This is a client-calculated field.</p>
Sync Local Size	<p>Values: number</p> <p>Internal Identifier: SyncSize</p> <p>A field used to determine status. The size of the working file in bytes. This is a client-calculated field.</p>
Sync Local Time Stamp	<p>Values: date/time</p> <p>Internal Identifier: SyncTime</p> <p>A field used to determine status. The time stamp for the working file. This is a client-calculated field.</p>
Sync MD5	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. The StarTeam client displays only significant zeroes, so 08, 0B, and 06 would become just 8, B, and 6, and A-F as a-f.</p> <p>Internal Identifier: SyncMD5</p> <p>Can not be used in queries. A field used to determine status. The MD5 checksum of the working file. This is a client-calculated field.</p>
Sync On Path To Root	Values: No , Yes

	<p>Internal Identifier: SyncOnPathToRoot</p> <p>A field used to determine status. When the working file is not based on the tip revision, this field indicates whether the server knows the relationship between the two. A Yes value in this field means that the working file needs to be merged or is out of date. A No value means that the relationship cannot be determined. This is a client-calculated field.</p>
Vault Branch Version (Advanced)	<p>Values: number</p> <p>Internal Identifier: VaultVersion</p> <p>The number of times a file has been checked in from the current view.</p>
Version (Advanced)	<p>Values: number</p> <p>Internal Identifier: RevisionNumber</p> <p>The last number in the branch revision number. For example, if the branch revision number is 1.3.1.2, the version is 2.</p>
View	<p>Values: list of views, <None></p> <p>Internal Identifier: ViewID</p> <p>The name of the view in which the item last branched. For example, if a file is inherited from a parent view but is branched in a child view, the value of this field in the child view changes from the name of the parent view to the name of the child view for the revision that branched and subsequent revisions in the child view.</p>
Working File Executable	<p>Values: text</p> <p>Internal Identifier: No, Yes</p> <p>Indicates whether the working file is executable. This is a client-calculated field.</p>
Working File Exists	<p>Values: No, Yes</p> <p>Internal Identifier: LocalFileExists</p> <p>Indicates whether a copy of a file is in its working folder. This is a client-calculated field.</p>
Working File Size	<p>Values: number</p> <p>Internal Identifier: LocalSize</p> <p>The size of the working file. This is a client-calculated field.</p>
Working File Time Stamp	<p>Values: date/time</p> <p>Internal Identifier: LocalTimestamp</p> <p>The time stamp of the working file. This is a client-calculated field.</p>

Related Reference

[Fields](#)
[Change Request Fields](#)
[Requirement Fields](#)
[Task Fields](#)
[Topic Fields](#)
[Folder Fields](#)
[Audit Fields](#)

Change Request Fields

This section lists all the change request fields in alphabetical order.

Note: Client-calculated fields cannot be used in custom email notifications or Notification Agent. Reports can use any field name.

Field	Description
Addressed By	Values: list of users, <None> Internal Identifier: AddressedBy Indicates the user who resolved a change request (resolved statuses are Cannot Reproduce, As Designed, Fixed, Documented, and Is Duplicate).
Addressed In	Values: list of view labels, <None> Internal Identifier: AddressedIn Indicates the next build label created and applied to the view after the resolution to a change request occurs.
Addressed In View	Values: list of views, <None> Internal Identifier: AddressedInView Indicates in what view the change request has been resolved. This is important for shared, and perhaps moved, change requests.
Attachment Count	Values: number Internal Identifier: AttachmentCount The number of files attached to a change request.
Attachment IDs (Advanced)	Values: Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [00 00 00 00 02 00 00 00] indicates two specific attachments. Internal Identifier: AttachmentIDs Cannot be used in queries. The ID numbers assigned to attachments. For example, the first attachment within a project is 00 00 00 00.
Attachment Names	Values: text containing a series of file names separated by spaces Internal Identifier: AttachmentNames The names of the files attached to a change request.
Branch On Change (Advanced)	Values: No, Yes Internal Identifier: BranchOnChange Indicates whether a change request will branch when it changes. The value is No if the change request's behavior is not set to "Branch On Change." Reasons for this may be: <ul style="list-style-type: none">– The change request is in the root or a reference view and the "Branch On Change" feature is disabled.– The change request is in a branching view but has already branched as a result of a change, which, in turn, results in the "Branch On Change" feature becoming disabled.– The change request is in a branching view, but its behavior currently does not permit it to branch on change. This means that modifications are checked into the parent view.

Note: If the value is [No](#), the value of the **Branch State** explains the [No](#).

The value [Yes](#) indicates that the change request resides in a branching view, has its behavior set to **Branch On Change**, but has yet to be changed.

Branch State (Advanced)

Values: [Branched](#), [Not Branched](#), [Root](#)

Internal Identifier: [BranchState](#)

Indicates whether a change request has branched in the child view, is still unbranched and, therefore, a part of the parent view, or was created in the view in which it resides.

The values [Branched](#) and [Not Branched](#) apply to change requests in branching views. The value [Root](#) applies to files created in the view in which the change request currently resides.

If the view is a reference view, it reflects the state of the change request in the reference view's parent.

Category

Values: text

Internal Identifier: [Category](#)

Text identifying the subcomponent in which the defect occurs. It is usually used in combination with the Component field.

Closed On

Values: date/time

Internal Identifier: [ClosedOn](#)

The date and time at which a change request was closed.

Comment

Values: text

Internal Identifier: [Comment](#)

The initial 2000 characters provided as the reason for changing a change request's properties are stored in the **Short Comment** field. The Comment field stores those 2000 characters and any additional text. Changing a change request's properties causes the application to create a new revision. Note: To include a Link comment, the Comment field is the value to use in an HTML Report.

CommentID (Advanced)

Values: number

Internal Identifier: [CommentID](#)

The ID number assigned to revision comment. Displays -1 if no revision comment was supplied.

Component

Values: text

Internal Identifier: [Component](#)

Text identifying the component in which the defect occurs. It is often used with the **Category** field to narrow that identification to a subcomponent.

Configuration Time

Values: date/time

Internal Identifier: [ConfigurationTime](#)

Indicates the time to which a change request is configured. If you configure a change request to a specific time, this field contains that time. If you configure a change request to a label or promotion state, this field shows either the time at which the label was created or the time at which the label associated with the promotion state was created.

CR Number

Values: number

	<p>Internal Identifier: ChangeNumber</p> <p>The number assigned to a change request. For example, if the Object ID is 0, the change request number is 1.</p>
Created By (Advanced)	<p>Values: list of users, <None></p> <p>Internal Identifier: CreatedUserID</p> <p>The name of the user who created the first revision in the view. This is either the user who initiated the change request or the user who modified the revision that branched.</p>
Created Time (Advanced)	<p>Values: date/time</p> <p>Internal Identifier: CreatedTime</p> <p>The time at which the first revision in the view was created.</p>
Deleted By	<p>Values: list of users, <None></p> <p>Internal Identifier: DeletedUserID</p> <p>The name of the user who deleted a change request. Because deleted change requests do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: DeletedTime</p> <p>The time at which a change request was deleted. Because deleted change requests do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Description	<p>Values: text</p> <p>Internal Identifier: Description</p> <p>The text in the Description field.</p>
Dot Notation	<p>Values: text</p> <p>Internal Identifier: DotNotation</p> <p>The ID assigned to a particular branch revision number. For example, if a change request was added to the current view (as opposed to inherited by the current view), its branch revision number is 1.x and its branch revision ID is 0. If a change request was branched in the current view, its branch revision ID is dependent on the revision number in the parent view and the number of IDs already assigned in the current view. For example, if a change request's revision number in the parent view is 1.7 at the time of the branch, and another change request with that same parent revision number was given the Branch Revision ID 6, this change request will also be given the Branch Revision ID 6.</p>
End Modified Time (Advanced)	<p>Values: date/time</p> <p>Internal Identifier: EndModifiedTime</p> <p>The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.</p>
Entered By	<p>Values: list of users, <None></p> <p>Internal Identifier: EnteredBy</p> <p>The name of the user who created this change request.</p>
Entered On	<p>Values: date/time</p>

	Internal Identifier: <code>EnteredOn</code>
External Reference	<p>The time at which this change request was created.</p> <p>Values: text</p> <p>Internal Identifier: <code>ExternalReference</code></p> <p>Text usually used to indicate a customer or other outside source who provided the data for this change request.</p>
Fix	<p>Values: text</p> <p>Internal Identifier: <code>Fix</code></p> <p>The text in the Fix field.</p>
Flag	<p>Values: <code>No</code>, <code>Yes</code></p> <p>Internal Identifier: <code>Flag</code></p> <p>A flag specifically marks/bookmarks change requests in the upper pane on your workstation. This is a client-calculated field.</p>
Flag User List (Advanced)	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, <code>[14 00 00 00]</code> indicates a specific user.</p> <p>Internal Identifier: <code>FlagUserList</code></p> <p>Cannot be used in queries. Identifies users who have set flags on a given item.</p>
Folder	<p>Values: text</p> <p>Internal Identifier: <code>Folder</code></p> <p>The name of the folder that stores the change request. This is a client-calculated field.</p>
Folder Path	<p>Values: text</p> <p>Internal Identifier: <code>Folder Path</code> (contains spaces)</p> <p>The path to the folder that stores the change request. This is a client-calculated field.</p>
Item Deleted By	<p>Values: list of users, <code>None</code></p> <p>Internal Identifier: <code>ItemDeletedUserID</code></p> <p>The name of the user who deleted this item. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Item Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: <code>ItemDeltedTime</code></p> <p>The time at which the item was created. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Last Build Tested	<p>Values: list of view labels, <code><None></code></p> <p>Internal Identifier: <code>LastBuildTested</code></p> <p>The build label selected by a user to represent the last build in which a change request was tested.</p>
Locked By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>ExclusiveLocker</code></p> <p>The name of user who has exclusively locked a change request.</p>
Modified By	<p>Values: list of users, <code><None></code></p>

	Internal Identifier: <code>ModifiedUserID</code>
Modified Time	<p>The name of the user who last modified a change request.</p> <p>Values: date/time</p> <p>Internal Identifier: <code>ModifiedTime</code></p> <p>The time at which a change request was last modified.</p>
My Lock	<p>Values: <code>Exclusively Locked By Me</code>, <code>Non-exclusively Locked By Me</code>, <code>Not Locked By Me</code></p> <p>Internal Identifier: <code>MyLock</code></p> <p>Indicates whether the current user has the change request locked and, if so, whether that lock is exclusive or not. This is a client-calculated field.</p>
New Revision Comment (Advanced)	<p>Values: text</p> <p>Internal Identifier: <code>NewRevisionComment</code></p> <p>Internal use only. The client uses this value during the item update process. The field always appears empty if added to the upper pane. This is a client-calculated field.</p>
Non-Exclusive Lockers	<p>Values: text</p> <p>Internal Identifier: <code>NonExclusiveLockers</code></p> <p>The names of the users who have locked the change request non-exclusively.</p>
Object ID	<p>Values: number</p> <p>Internal Identifier: <code>ID</code></p> <p>Each change request is assigned an object ID when it is added to a view. When it is branched in a child view, it is assigned another object ID. The original ID belongs to the change request in the parent view.</p>
Parent Branch Revision (Advanced)	<p>Values: number</p> <p>Internal Identifier: <code>PathRevision</code></p> <p>The last number in the branch revision number before a change request branched. For example, if this number is <code>7</code>, the branch revision was <code>1.7</code> at the time the change request branched (becoming <code>1.7.1.0</code>, as seen in the change request's history). This number is <code>-1</code> if a change request was not inherited from the parent view.</p>
Parent ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: <code>ParentObjectID</code></p> <p>The object ID of a change request in the parent view. The Parent ID is <code>-1</code> if this view has no parent view.</p>
Parent Revision (Advanced)	<p>Values: number</p> <p>Internal Identifier: <code>PathRevision</code></p> <p>The revision number at which a change request branched. For example, if this number is <code>8</code>, this change request's revision number in the parent view was <code>8</code> at the time the change request branched. The history should show that revision <code>9</code> in the first revision in the current view. This number is <code>0</code> if this change request was not inherited from the parent view.</p>
Platform	<p>Values: <code>All</code>, <code>MacOS</code>, <code>Other</code>, <code>Unix</code>, <code>Windows 2000</code>, <code>Windows 95</code>, <code>Windows 98</code>, <code>Windows NT</code>, <code>Windows XP</code></p>

	Internal Identifier: Platform
	The value of the Platform field.
Priority	Values: No , Yes
	Internal Identifier: Priority
	The value of the Priority field. Many people use repository customization to extend this field to include other values because Booleans in the application are treated as enumerated types. For example, No is 0 and Yes is 1. An administrator might change No to Not A Priority , Yes to Priority 1 , and add Priority 2 through Priority 10 .
Read Only (Advanced)	Values: No , Yes
	Internal Identifier: ReadOnly
	Indicates whether the change request's configuration is read-only (as in a rollback configuration of a view) and/or its behavior does not allow it to branch on modification.
Read Status	Values: Read , Unread
	Internal Identifier: ReadStatus
	Indicates whether a change request is considered read or not read. This is a client-calculated field.
Read Status User List	Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.
	Internal Identifier: ReadStatusUserList
	Can not be used in queries. Identifies users for whom a given item's status is Unread .
Resolved On	Values: date/time
	Internal Identifier: ResolvedOn
	The time at which a change request was resolved. The resolution can be: Cannot Reproduce , As Designed , Fixed , Documented , or Is Duplicate .
Responsibility	Values: list of users, <None>
	Internal Identifier: Responsibility
	The name of the user who is currently responsible for a change request.
Revision Flags (Advanced)	Values: 0
	Internal Identifier: RevisionFlags
	Internal use only.
Root Object ID (Advanced)	Values: number
	Internal Identifier: RootObjectID
	The object ID of the oldest ancestor of a change request. For example, if a change request was not inherited from a parent view, the root object ID is the same as its object ID. If it was inherited from a parent view, the root object ID is the Parent ID, or the parent's Parent ID.
Severity	Values: High , Low , Medium
	Internal Identifier: Severity
	The value of the Severity field.
Share State	Values: DerivedShare , Not Shared , Root Share

	<p>Internal Identifier: ShareState</p> <p>Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.</p>
Short Comment	<p>Values: text</p> <p>Internal Identifier: ShortComment</p> <p>Stores the initial 2000 characters provided as the reason for changing a change request's properties. Additional text is stored in the Comment field.</p>
Status	<p>Values: New, Open, In Progress, Deferred, Cannot Reproduce, As Designed, Fixed, Documented, Is Duplicate, Verified Deferred, Verified Cannot Reproduce, Verified As Designed, Verified Fixed, Verified Documented, Verified Is Duplicate, Closed Deferred, Closed Cannot Reproduce, Closed As Designed, Closed Fixed, Closed Documented, Closed Is Duplicate</p> <p>Internal Identifier: Status</p> <p>The value of the Status field.</p>
Synopsis	<p>Values: text</p> <p>Internal Identifier: Synopsis</p> <p>The value of the Synopsis field.</p>
Test Command	<p>Values: text</p> <p>Internal Identifier: TestCommand</p> <p>The text in the Test Command field.</p>
Type	<p>Values: Defect, Suggestion</p> <p>Internal Identifier: Type</p> <p>The value of the Type field.</p>
Verified On	<p>Values: date/time</p> <p>Internal Identifier: VerifiedOn</p> <p>The time at which a change request was verified. The resolution can be Verified Cannot Reproduce, Verified As Designed, Verified Fixed, Verified Documented, or Verified Is Duplicate.</p>
Version (Advanced)	<p>Values: number</p> <p>Internal Identifier: RevisionNumber</p> <p>The last number in the branch revision number. For example, if the branch revision number is 1.3.1.2, the version is 2.</p>
View	<p>Values: list of views, <None></p> <p>Internal Identifier: ViewID</p> <p>The name of the view in which the item last branched. For example, if a change request is inherited from a parent view but is branched in the child view, the value of this field in the child view changes from the name of the parent view to the name of the child view for the revision that branched and subsequent revisions in the child view.</p>
Work Around	<p>Values: text</p>

Internal Identifier: [WorkAround](#)

The text in the **Work Around** field.

Related Reference

[Fields](#)

[File Fields](#)

[Requirement Fields](#)

[Task Fields](#)

[Topic Fields](#)

[Folder Fields](#)

[Audit Fields](#)

Requirement Fields

This section lists all the requirement fields in alphabetical order.

Note: Client-calculated fields cannot be used in custom email notifications or Notification Agent. Reports can use any field name.

Field	Description
Am I Responsible?	Values: <code>No</code> , <code>Yes</code> Internal Identifier: <code>AmIResponsible</code> Indicates whether the logged-on user is responsible for a requirement. This is a client-calculated field.
Ambiguities Found	Values: number Internal Identifier: <code>AmbiguitiesFound</code> Indicates the number of ambiguities found in the requirement.
Attachment Count	Values: Internal Identifier: Values: number Internal Identifier: <code>AttachmentCount</code> The number of files attached to a requirement.
Attachment IDs (Advanced)	Values: byte array; displayed as a bracketed series of numbers in hex format. For example, <code>[00 00 00 00 02 00 00 00]</code> indicates two specific attachments. Internal Identifier: Cannot be used in queries. The ID numbers assigned to attachments. For example, the first attachment within a project is <code>00 00 00 00</code> .
Attachment Names	Values: Internal Identifier: Values: text containing a series of file names separated by spaces Internal Identifier: <code>AttachmentNames</code> The names of the files attached to a requirement.
Children Count	Values: number Internal Identifier: <code>ChildrenCount</code> The number of responses that are children of this requirement. This is a client-calculated field.
ChildType	Values: <code>Child Requirement</code> , <code>Requirement</code> Internal Identifier: <code>ChildType</code> Indicates whether the requirement is the root of a requirement tree or a child of another requirement. This is a client-calculated field.
Comment	Values: text Internal Identifier: <code>Comment</code> The initial 2000 characters provided as the reason for changing a requirement's properties are stored in the Short Comment field. This field, the Comment field, stores those 2000 characters and any additional text. Changing a requirement's properties causes the application to create a new

	revision. Note: To include a Link comment, the Comment field is the value to use in an HTML Report.
CommentID (Advanced)	<p>Values: number</p> <p>Internal Identifier: CommentID</p> <p>The ID number assigned to a revision comment. Displays -1 if no revision comment was supplied.</p>
Comments	<p>Values: text</p> <p>Internal Identifier: Comments</p> <p>Provides comments about the revised description created because of ambiguities found in the original description or for other reasons.</p>
Configuration Time	<p>Values: date/time</p> <p>Internal Identifier: ConfigurationTime</p> <p>Indicates the time to which a requirement is configured. If you configure a requirement to a specific time, this field contains that time. If you configure a requirement to a label or promotion state, this field shows either the time at which the label was created or the time at which the label associated with the promotion state was created.</p>
Created By	<p>Values: list of users, <None></p> <p>Internal Identifier: CreadedUserID</p> <p>The name of the user who created the first revision in the view. This is the user who initiated the requirement.</p>
Created Time	<p>Values: date/time</p> <p>Internal Identifier: CreatedTime</p> <p>The time at which the first revision in the view was created.</p>
Deleted By	<p>Values: list of users, <None></p> <p>Internal Identifier: DeletedUserID</p> <p>The name of the user who deleted a requirement. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: DeletedTime</p> <p>The time at which a requirement was deleted. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Description	<p>Values: text</p> <p>Internal Identifier: Description</p> <p>The text in the Description field.</p>
Disabled	<p>Values: No, Yes</p> <p>Internal Identifier: Disabled</p> <p>Indicates whether the requirement is disabled.</p>
Dot Notation	<p>Values: text</p> <p>Internal Identifier: DotNotation</p> <p>The branch revision number, for example, 1.2.</p>
End Modified Time (Advanced)	<p>Values: date/time</p>

	<p>Internal Identifier: EndModifiedTime</p> <p>The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.</p>
Expected Effort	<p>Values: number</p> <p>Internal Identifier: ExpectedEffort</p> <p>Indicates the expected case estimate for how long it will take to implement the requirement fully. If you are publishing requirements from CaliberRM to StarTeam, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Low Effort and the High Effort fields, and should be used consistently for all requirements.</p>
External Reference	<p>Values: text</p> <p>Internal Identifier: ExternalReference</p> <p>Usually provides the name of an external customer who asked for this requirement.</p>
Flag	<p>Values: No, Yes</p> <p>Internal Identifier: Flag</p> <p>A flag specifically marks/bookmarks requirements in the upper pane on your workstation. This is a client-calculated field.</p>
Flag User List (Advanced)	<p>Values: byte array; displayed as a bracketed series numbers in hex format. For example, [14 00 00 00] indicates a specific user</p> <p>Internal Identifier: FlagUserList</p> <p>Can not be used in queries. Identifies users who have set flags on a given item.</p>
Folder Path	<p>Values: text</p> <p>Internal Identifier: Folder Path (contains spaces)</p> <p>The path to the folder that stores the requirement. This is a client-calculated field.</p>
High Effort	<p>Values: number</p> <p>Internal Identifier: HighEffort</p> <p>Indicates the worst case estimate for how long it will take to implement the requirement fully. If you are publishing requirements from CaliberRM to StarTeam, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Low Effort and the Expected Effort fields, and should be used consistently for all requirements.</p>
Item Deleted By	<p>Values: list of users, None</p> <p>Internal Identifier: ItemDeletedUserID</p> <p>The name of the user who deleted this item. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Item Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: ItemDeltedTime</p> <p>The time at which the item was created. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>

Locked By	<p>Values: list of users, <None></p> <p>Internal Identifier: <code>ExclusiveLocker</code></p> <p>The name of the user who has exclusively locked a requirement.</p>
Low Effort	<p>Values: number</p> <p>Internal Identifier: <code>LowEffort</code></p> <p>Indicates the best case estimate for how long it will take to implement the requirement fully. If you are publishing requirements from CaliberRM to StarTeam, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Expected Effort and the High Effort fields, and should be used consistently for all requirements.</p>
Modified By	<p>Values: list of users, <None></p> <p>Internal Identifier: <code>ModifiedUserID</code></p> <p>The name of the user who last modified a requirement.</p>
Modified Time	<p>Values: date/time</p> <p>Internal Identifier: <code>ModifiedTime</code></p> <p>The time at which a requirement was last modified.</p>
My Lock	<p>Values: <code>Exclusively Locked By Me</code>, <code>Non-exclusively Locked By Me</code>, <code>Not Locked By Me</code></p> <p>Internal Identifier: <code>MyLock</code></p> <p>Indicates whether the current user has the requirement locked and, if so, whether that lock is exclusive or not. This is a client-calculated field.</p>
Name	<p>Values: text</p> <p>Internal Identifier: <code>Name</code></p> <p>The name of the requirement.</p>
New Revision Comment (Advanced)	<p>Values: text</p> <p>Internal Identifier: <code>NewRevisionComment</code></p> <p>Internal use only. A client uses this value during the item update process. The field always appears empty if added to the upper pane. This is a client-calculated field.</p>
Non-Exclusive Lockers	<p>Values: text</p> <p>Internal Identifier: <code>NonExclusiveLockers</code></p> <p>The names of the users who have locked the requirement non-exclusively.</p>
Notes	<p>Values: text</p> <p>Internal Identifier: <code>Notes</code></p> <p>Text comments on the effort levels for this requirement.</p>
Number	<p>Values: number</p> <p>Internal Identifier: <code>RequirementNumber</code></p> <p>Number identifying the requirement. For example, if the Object ID is 0, the requirement number is 1.</p>
Object ID	<p>Values: number</p> <p>Internal Identifier: <code>ID</code></p> <p>Each requirement is assigned an object ID when it is added to a view.</p>

Owner	<p>Values: list of users, <None></p> <p>Internal Identifier: Owner</p> <p>Indicates who is ultimately responsible for this requirement.</p>
Parent Requirement ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: ParentRequirementID</p> <p>The object ID of a requirement in the parent view. The Parent ID is -1 if this view has no parent view.</p>
Priority	<p>Values: Desirable, Essential, Unassigned, Useful</p> <p>Internal Identifier: Priority</p> <p>The value of the Priority field. You can use repository customization to change the names of these values or include other values.</p>
Read Only (Advanced)	<p>Values: No, Yes</p> <p>Internal Identifier: ReadOnly</p> <p>Indicates whether the requirement's configuration is read-only (as in a rollback configuration of a view).</p>
Read Status	<p>Values: Read, Unread</p> <p>Internal Identifier: ReadStatus</p> <p>Indicates whether a requirement is considered read or not read. This is a client-calculated field.</p>
Read Status User List	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.</p> <p>Internal Identifier: ReadStatusUserList</p> <p>Can not be used in queries. Identifies users for whom a given item's status is Unread.</p>
Responsible Count	<p>Values: number</p> <p>Internal Identifier: ResponsibleCount</p> <p>The number of users who are responsible for a requirement.</p>
Responsible IDs	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.</p> <p>Internal Identifier: ResponsibleIDs</p> <p>Can not be used in queries. The ID numbers assigned to the users who are responsible for the requirement.</p>
Responsible Names	<p>Values: text containing a series of user names separated by spaces</p> <p>Internal Identifier: ResponsibleNames</p> <p>The names of the users responsible for this requirement.</p>
Reviewed By	<p>Values: byte array</p> <p>Internal Identifier: ReviewedByIDs</p> <p>Can not be used in queries. Should not be used at all.</p>
Revised Description	<p>Values: text</p> <p>Internal Identifier: RevisedDescription</p>

	Provides a new, revised description because of ambiguities found in the original description or for other reasons.
Revision Flags (Advanced)	Values: 0 Internal Identifier: RevisionFlags
Share State	Internal use only. Values: DerivedShare , Not Shared , Root Share Internal Identifier: ShareState Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.
Short Comment	Values: text Internal Identifier: ShortComment
Status	Stores the initial 2000 characters provided as the reason for changing a requirement's properties. Additional text is stored in the Comment field. Values: Accepted , Approved , Complete , Deferred , Draft , Pending , ReadyForCCB , Rejected , Review , Submitted Internal Identifier: Status
Type	Indicates the status of this requirement. Values: Business Requirement , Business Specification , Hardware Requirement , Hardware Specification , Human Resources , Information Technology , Software Requirement , Software Specification Internal Identifier: Type
Version (Advanced)	Indicates the type of requirement. Values: number Internal Identifier: RevisionNumber The last number in the branch revision number. For example, if the branch revision number is 1.2 , the version is 2 .

Related Reference

[Fields](#)
[File Fields](#)
[Change Request Fields](#)
[Task Fields](#)
[Topic Fields](#)
[Folder Fields](#)
[Audit Fields](#)

Task Fields

This section lists all the task fields in alphabetical order.

Note: Client-calculated fields cannot be used in custom email notifications or Notification Agent. Reports can use any field name.

Field	Description
Actual Finish	<p>Values: date/time</p> <p>Internal Identifier: <code>StTaskActualFinish</code></p> <p>The actual finish date for a task.</p>
Actual Hours	<p>Values: number</p> <p>Internal Identifier: <code>StTaskActualHours</code></p> <p>The number of hours spent completing the task.</p>
Actual Start	<p>Values: date/time</p> <p>Internal Identifier: <code>StTaskActualStart</code></p> <p>The actual start date for a task.</p>
Attachment Count	<p>Values: number</p> <p>Internal Identifier: <code>AttachmentCount</code></p> <p>The number of files attached to a task.</p>
Attachment IDs (Advanced)	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, <code>[00 00 00 00 02 00 00 00]</code> indicates two specific attachments.</p> <p>Internal Identifier: <code>AttachmentIDs</code></p> <p>Can not be used in queries. The ID numbers assigned to attachments. For example, the first attachment within a project is <code>00 00 00 00</code>.</p>
Attachment Names	<p>Values: text containing a series of file names separated by spaces</p> <p>Internal Identifier: <code>AttachmentNames</code></p> <p>The names of the files attached to a task.</p>
Attention Notes	<p>Values: text</p> <p>Internal Identifier: <code>StTaskAttentionNotes</code></p> <p>The text in the Needs Attention note.</p>
Children Count	<p>Values: number</p> <p>Internal Identifier: <code>ChildrenCount</code></p> <p>The number of responses that are subtasks of this task. This is a client-calculated field.</p>
Comment	<p>Values: text</p> <p>Internal Identifier: <code>Comment</code></p> <p>The initial 2000 characters provided as the reason for changing a change request's properties are stored in the Short Comment field. This field, the Comment field, stores those 2000 characters and any additional text. Changing a task's properties causes the application to create a new revision. Note: To include a Link comment, the Comment field is the value to use in an HTML Report.</p>
CommentID (Advanced)	<p>Values: number</p>

	<p>Internal Identifier: <code>CommentID</code></p> <p>The ID number assigned to revision comment. Displays <code>-1</code> if no revision comment was supplied.</p>
Configuration Time	<p>Values: date/time</p> <p>Internal Identifier: <code>ConfigurationTime</code></p> <p>Indicates the time to which a task is configured. If you configure a task to a specific time, this field contains that time. If you configure a task to a label or promotion state, this field shows either the time at which the label was created or the time at which the label associated with the promotion state was created.</p>
Constraint Date	<p>Values: date/time</p> <p>Internal Identifier: <code>StTaskConstraintDate</code></p> <p>A task's constraint date from MS Project.</p>
Constraint Type	<p>Values: <code>As Late As Possible</code>, <code>As Soon As Possible</code>, <code>Finish No Earlier Than</code>, <code>Finish No Later Than</code>, <code>Must Finish On</code>, <code>Must Start On</code>, <code>Start No Earlier Than</code>, <code>Start No Later Than</code></p> <p>Internal Identifier: <code>StTaskConstraintType</code></p> <p>A task's constraint type from MS Project.</p>
Created By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>StTaskConstraintType</code></p> <p>The name of the user who created the first revision in the view. This is the user who initiated the task.</p>
Created Time	<p>Values: date/time</p> <p>Internal Identifier: <code>CreatedTime</code></p> <p>The time at which the first revision in the view was created.</p>
Deleted By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>DeletedUserID</code></p> <p>The name of the user who deleted a task. Because deleted tasks do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: <code>DeletedTime</code></p> <p>The time at which a task was deleted. Because deleted tasks do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Dot Notation	<p>Values: text</p> <p>Internal Identifier: <code>DotNotation</code></p> <p>The branch revision number, for example, <code>1.2</code>.</p>
End Modified Time (Advanced)	<p>Values: date/time</p> <p>Internal Identifier: <code>EndModifiedTime</code></p> <p>The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.</p>
Estimated Finish	<p>Values: date/time</p> <p>Internal Identifier: <code>StTaskEstimatedFinish</code></p>

Estimated Finish Variance	<p>The estimated finish date for a task.</p> <p>Values: date/time</p> <p>Internal Identifier: <code>StTaskEstimatedFinishVariance</code></p>
Estimated Hours	<p>The difference between the estimated and the actual finish date for a task.</p> <p>Values: number</p> <p>Internal Identifier: <code>StTaskEstimatedHours</code></p>
Estimated Hours Variance	<p>The number of hours spent completing the task.</p> <p>Values: number</p> <p>Internal Identifier: <code>StTaskEstimatedHoursVariance</code></p>
Estimated Start	<p>The difference between the estimated and the actual number of hours spent completing the task.</p> <p>Values:</p> <p>Internal Identifier:</p>
Estimated Start Variance	<p>Values: date/time</p> <p>Internal Identifier: <code>StTaskEstimatedStart</code></p> <p>The estimated start date for a task.</p>
Flag	<p>Values: date/time</p> <p>Internal Identifier: <code>StTaskEstimatedStartVariance</code></p> <p>The difference between the estimated and the actual start date for a task.</p>
Flag User List (Advanced)	<p>Values: No, Yes</p> <p>Internal Identifier: <code>Flag</code></p> <p>A flag specifically marks/bookmarks tasks in the upper pane on your workstation. This is a client-calculated field.</p>
Folder Path	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, <code>[14 00 00 00]</code> indicates a specific user.</p> <p>Internal Identifier: <code>FlagUserList</code></p>
Is My Task?	<p>Can not be used in queries. Identifies users who have set flags on a given item.</p> <p>Values: text</p> <p>Internal Identifier: <code>Folder Path</code> (contains spaces)</p>
Is Replicated	<p>The path to the folder that stores the task. This is a client-calculated field.</p> <p>Values: No, Yes</p> <p>Internal Identifier: <code>IsMyTask?</code></p>
Item Deleted By	<p>Indicates whether the logged on user is responsible for a task. This is a client-calculated field.</p> <p>Values: 0, 1</p> <p>Internal Identifier: (contains spaces)</p>
	<p>Indicates whether the task is from MS Project task.</p> <p>Values: list of users, None</p> <p>Internal Identifier: <code>ItemDeletedUserID</code></p>
	<p>The name of the user who deleted this item. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>

Item Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: ItemDeltedTime</p> <p>The time at which the item was created. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Last MS Project Update	<p>Values: date/time</p> <p>Internal Identifier: StTaskMSProjectLastUpdate</p> <p>The date that a task was last updated from MS Project.</p>
Last Work/Dependency Update	<p>Values: date/time</p> <p>Internal Identifier: StWorkDependencyLastUpdate</p> <p>The last time that a work record or a dependency (task successor or predecessor) was added, edited, or deleted. This field is for use with MS Project.</p>
Locked By	<p>Values: list of users, <None></p> <p>Internal Identifier: ExclusiveLocker</p> <p>The name of user who has exclusively locked a task.</p>
Milestone	<p>Values: No, Yes</p> <p>Internal Identifier: StTaskMilestone</p> <p>Indicates whether a task represents a milestone. In MS Project, the definition for a milestone is a task of zero time length. It serves as a heading for one or more tasks to which a time length has been assigned.</p> <p>In the application, a task has a milestone check box. After work is assigned to a task, it is no longer a milestone.</p>
Modified By	<p>Values: list of users, <None></p> <p>Internal Identifier: ModifiedUserID</p> <p>The name of the user who last modified a task.</p>
Modified Time	<p>Values: date/time</p> <p>Internal Identifier: ModifiedTime</p> <p>The time at which a task was last modified.</p>
MS Project File Name (Advanced)	<p>Values: text</p> <p>Internal Identifier: StTaskMSProjectFileName</p> <p>The name of the MS project file from which a task was exported.</p>
MS Task GUID (Advanced)	<p>Values: text</p> <p>Internal Identifier: StTaskGUID</p> <p>The GUID for a task in MS Project.</p>
MS Task Unique ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: StTaskUniqueID</p> <p>The unique ID for a task in MS Project.</p>
MS WBS Code (Advanced)	<p>Values: text</p> <p>Internal Identifier: StTaskWBSCode</p> <p>A task's WBS code from MS Project.</p>
My Lock	<p>Values: Exclusively Locked By Me, Non-exclusively Locked By Me, Not Locked By Me</p>

	Internal Identifier: <code>MyLock</code>
	Indicates whether the current user has the task locked and, if so, whether that lock is exclusive or not. This is a client-calculated field.
Needs Attention	Values: <code>No</code> , <code>Yes</code>
	Internal Identifier: <code>StTaskNeedsAttention</code>
	Indicates that the check box for Needs Attention has been selected.
New Revision Comment (Advanced)	Values: text
	Internal Identifier: <code>NewRevisionComment</code>
	Internal use only. The client uses this value during the item update process. The field always appears empty if added to the upper pane. This is a client-calculated field.
Non-Exclusive Lockers	Values: text
	Internal Identifier: <code>NonExclusiveLockers</code>
	The names of the users who have locked the task non-exclusively.
Notes	Values: text
	Internal Identifier: <code>StTaskNotes</code>
	The text of the note that accompanies the Needs Attention field.
Object ID	Values: number
	Internal Identifier: <code>ID</code>
	Each task is assigned an object ID when it is added to a view.
Parent Task ID (Advanced)	Values: number
	Internal Identifier: <code>StTaskParentID</code>
	The object ID of a task in the parent view. The Parent ID is -1 if this view has no parent view.
Percent Complete	Values: number
	Internal Identifier: <code>StTaskPercentComplete</code>
	A percentage indicating how much of a task has been completed.
Priority	Values: <code>Do Not Level</code> , <code>High</code> , <code>Higher</code> , <code>Highest</code> , <code>Low</code> , <code>Lower</code> , <code>Lowest</code> , <code>Medium</code> , <code>Very High</code> , <code>Very Low</code>
	Internal Identifier: <code>StTaskPriority</code>
	Indicates the priority given to a task. These priorities are identical to those in MS Project.
Read Only (Advanced)	Values: <code>No</code> , <code>Yes</code>
	Internal Identifier: <code>ReadOnly</code>
	Indicates whether the task's configuration is read-only (as in a rollback configuration of a view).
Read Status	Values: <code>Read</code> , <code>Unread</code>
	Internal Identifier: <code>ReadStatus</code>
	Indicates whether a task is considered read or not read. This is a client-calculated field.
Read Status User List	Values: byte array; displayed as a bracketed series of numbers in hex format. For example, <code>[14 00 00 00]</code> indicates a specific user.

	<p>Internal Identifier: ReadStatusUserList</p> <p>Can not be used in queries. Identifies users for whom a given item's status is Unread.</p>
Resource Count	<p>Values: number</p> <p>Internal Identifier: StTaskResourceCount</p> <p>The number of users listed as resources for a task.</p>
Resource IDs (Advanced)	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.</p> <p>Internal Identifier: StTaskResourceIDs</p> <p>Can not be used in queries. The ID numbers assigned to the users who are this task's resources.</p>
Resource Names	<p>Values: text containing a series of user names separated by spaces</p> <p>Internal Identifier: StTaskResourceNames</p> <p>The names of the users who are this task's resources.</p>
Responsibility	<p>Values: list of users, <None></p> <p>Internal Identifier: StTaskResponsibility</p> <p>The name of the user who is currently responsible for the task.</p>
Revision Flags (Advanced)	<p>Values: 0</p> <p>Internal Identifier: RevisionFlags</p> <p>Internal use only.</p>
Share State	<p>Values: DerivedShare, Not Shared, Root Share</p> <p>Internal Identifier: ShareState</p> <p>Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.</p>
Short Comment	<p>Values: text</p> <p>Internal Identifier: ShortComment</p> <p>Stores the initial 2000 characters provided as the reason for changing a task's properties. Additional characters are stored in the Comment field.</p>
Status	<p>Values: Closed, Finish, Hold, In Progress, Pending, Ready To Start</p> <p>Internal Identifier: STTaskStatus</p> <p>Indicates the status of the task.</p>
Task Duration	<p>Values: number</p> <p>Internal Identifier: STTaskDuration</p> <p>The number of hours during which any user is working on a task. For example if two people will work eight hours on a task, the duration is eight hours if they work at the same time or a maximum of 16 hours if they do the work on different days.</p>
Task Name	<p>Values: text</p> <p>Internal Identifier: STTaskName</p> <p>The name of the task.</p>

Task Number	<p>Values: number</p> <p>Internal Identifier: StTaskNumber</p> <p>The number assigned to a task. For example, if the Object ID is 0, the task number is 1.</p>
Task Origin	<p>Values: MSProject, StarTeam</p> <p>Internal Identifier: STTaskOrigin</p> <p>Indicates whether the task was created in the application or exported to the application from Microsoft Project.</p>
Task Type	<p>Values: Fixed Duration, Fixed Units, Fixed Work</p> <p>Internal Identifier: StTaskType</p> <p>A task's type in MS Project.</p>
Version (Advanced)	<p>Values: number</p> <p>Internal Identifier: RevisionNumber</p> <p>The last number in the branch revision number. For example, if the branch revision number is 1.2, the version is 2.</p>
Work Record Count	<p>Values: number</p> <p>Internal Identifier: WorkRecCount</p> <p>The number of work records currently added to a task.</p>

Related Reference

[Fields](#)
[File Fields](#)
[Change Request Fields](#)
[Requirement Fields](#)
[Topic Fields](#)
[Folder Fields](#)
[Audit Fields](#)

Topic Fields

This section lists all the topic fields in alphabetical order.

Note: Client-calculated fields cannot be used in custom email notifications or Notification Agent. Reports can use any field name.

Field	Description
Am I Recipient?	Values: No, Yes Internal Identifier: <code>AmIRecipient?</code> Indicates whether the logged on user is a recipient of a topic. This is a client-calculated field.
Attachment Count	Values: number Internal Identifier: <code>AttachmentCount</code> The number of files attached to a topic.
Attachment IDs (Advanced)	Values: byte array; displayed as a bracketed series of numbers in hex format. For example, <code>[00 00 00 00 02 00 00 00]</code> indicates two specific attachments. Internal Identifier: <code>AttachmentIDs</code> Can not be used in queries. The ID numbers assigned to attachments. For example, the first attachment within a project is <code>00 00 00 00</code> .
Attachment Names	Values: text containing a series of file names separated by spaces Internal Identifier: <code>AttachmentNames</code> The names of the files attached to a topic.
Children Count	Values: number Internal Identifier: <code>ChildrenCount</code> The number of responses that are children of this topic. This is a client-calculated field.
Comment	Values: text Internal Identifier: <code>Comment</code> The initial 2000 characters provided as the reason for changing a topic's properties are stored in the Short Comment field. This field, the Comment field, stores those 2000 characters and any additional text. Changing a topic's properties causes the application to create a new revision. Note: To include a Link comment, the Comment field is the value to use in an HTML Report.
CommentID (Advanced)	Values: number Internal Identifier: <code>CommentID</code> The ID number assigned to revision comment. Displays <code>-1</code> if no revision comment was supplied.
Configuration Time	Values: date/time Internal Identifier: <code>ConfigurationTime</code> Indicates the time to which a topic is configured. If you configure a topic to a specific time, this field contains that time. If you configure a topic to a label or promotion state, this field shows either the time at which the label was

	created or the time at which the label associated with the promotion state was created.
Content	<p>Values: text</p> <p>Internal Identifier: <code>Description</code></p> <p>The text of a topic.</p>
Created By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>CreatedUserID</code></p> <p>The name of the user who created the first revision in the view. This is the user who initiated the topic.</p>
Created Time	<p>Values: date/time</p> <p>Internal Identifier: <code>CreatedTime</code></p> <p>The time at which the first revision in the view was created.</p>
Deleted By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>DeletedUserID</code></p> <p>The name of the user who deleted a topic. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: <code>DeletedTime</code></p> <p>The time at which a topic was deleted. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Dot Notation	<p>Values: text</p> <p>Internal Identifier: <code>DotNotation</code></p> <p>The branch revision number, for example, <code>1.2</code>.</p>
End Modified Time (Advanced)	<p>Values: date/time</p> <p>Internal Identifier: <code>EndModifiedTime</code></p> <p>The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.</p>
Flag	<p>Values: <code>No</code>, <code>Yes</code></p> <p>Internal Identifier: <code>Flag</code></p> <p>A flag specifically marks/bookmarks topics and/or responses in the upper pane on your workstation. This is a client-calculated field.</p>
Flag User List (Advanced)	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, <code>[14 00 00 00]</code> indicates a specific user.</p> <p>Internal Identifier: <code>FlagUserList</code></p> <p>Can not be used in queries. Identifies users who have set flags on a given item.</p>
Folder Path	<p>Values: text</p> <p>Internal Identifier: <code>Folder Path</code> (contains spaces)</p> <p>The path to the folder that stores the topic. This is a client-calculated field.</p>
Item Deleted By	<p>Values: list of users, <code>None</code></p> <p>Internal Identifier: <code>ItemDeletedUserID</code></p>

Item Deleted Time	<p>The name of the user who deleted this item. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p> <p>Values: date/time</p> <p>Internal Identifier: <code>ItemDeltedTime</code></p>
Locked By	<p>The time at which the item was created. Because deleted items do not appear in the list, this information is unavailable to users. Internal Use Only.</p> <p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>ExclusiveLocker</code></p>
Modified By	<p>The name of the user who has exclusively locked a topic.</p> <p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>ModifiedUserID</code></p>
Modified Time	<p>The name of the user who last modified a topic.</p> <p>Values: date/time</p> <p>Internal Identifier: <code>ModifiedTime</code></p>
My Lock	<p>The time at which a topic was last modified.</p> <p>Values: <code>Exclusively Locked By Me</code>, <code>Non-exclusively Locked By Me</code>, <code>No Locked By Me</code></p> <p>Internal Identifier: <code>MyLock</code> (contains spaces)</p> <p>Indicates whether the current user has the topic locked and, if so, whether that lock is exclusive or not. This is a client-calculated field.</p>
New Revision Comment (Advanced)	<p>Values: text</p> <p>Internal Identifier: <code>NewRevisionComment</code></p> <p>Internal use only. the client uses this value during the item update process. The field always appears empty if added to the upper pane. This is a client-calculated field.</p>
Non-Exclusive Lockers	<p>Values: text</p> <p>Internal Identifier: <code>NonExclusiveLockers</code></p>
Object ID	<p>The names of the users who have locked the topic non-exclusively.</p> <p>Values: number</p> <p>Internal Identifier: <code>ID</code></p>
Parent Topic ID (Advanced)	<p>Each topic is assigned an object ID when it is added to a view.</p> <p>Values: number</p> <p>Internal Identifier: <code>ParentTopicID</code></p>
Priority	<p>The object ID of a topic in the parent view. The Parent ID is -1 if this view has no parent view.</p> <p>Values: <code>High</code>, <code>Low</code>, <code>Normal</code></p> <p>Internal Identifier: <code>Priority</code></p>
Read Only (Advanced)	<p>The value of the Priority field. You can use repository customization to change the names of these values or include other values.</p> <p>Values: <code>No</code>, <code>Yes</code></p> <p>Internal Identifier: <code>ReadOnly</code></p>

	Indicates whether the topic's configuration is read-only (as in a rollback configuration of a view).
Read Status	<p>Values: Read, Unread</p> <p>Internal Identifier: ReadStatus</p> <p>Indicates whether a topic is considered read or not read. This is a client-calculated field.</p>
Read Status User List	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.</p> <p>Internal Identifier: ReadStatusUserList</p> <p>Cannot be used in queries. Identifies users for whom a given item's status is Unread.</p>
Recipient Count	<p>Values: number</p> <p>Internal Identifier: RecipientCount</p> <p>The number of recipients to whom a topic is addressed.</p>
Recipient IDs	<p>Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.</p> <p>Internal Identifier: RecipientIDs</p> <p>Can not be used in queries. The ID numbers assigned to the users who are recipients (people to be notified about this topic).</p>
Recipient Names	<p>Values: text containing a series of users names separated by spaces</p> <p>Internal Identifier: Recipient Names</p> <p>The names of the recipients designated for notification about this topic.</p>
Revision Flags (Advanced)	<p>Values: 0</p> <p>Internal Identifier: RevisionFlags</p> <p>Internal use only.</p>
Share State	<p>Values: DerivedShare, Not Shared, Root Share</p> <p>Internal Identifier: ShareState</p> <p>Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.</p>
Short Comment	<p>Values: text</p> <p>Internal Identifier: ShortComment</p> <p>Stores the initial 2000 characters provided as the reason for changing a topic's properties. Additional text is stored in the Comment field.</p>
Status	<p>Values: Active, Inactive</p> <p>Internal Identifier: Status</p> <p>Indicates the status of this topic.</p>
Title	<p>Values: text</p> <p>Internal Identifier: Title</p> <p>The text of the Title field.</p>
Topic Number	<p>Values: number</p> <p>Internal Identifier: TopicNumber</p>

Type	<p>The number assigned to a topic. For example, if the Object ID is 0, the topic number is 1.</p> <p>Values: Response, Topic</p> <p>Internal Identifier: Type</p> <p>Indicates whether the item is a topic (root of a topic tree) or a response (branch of a topic tree). This is a client-calculated field.</p>
Version (Advanced)	<p>Values: number</p> <p>Internal Identifier: RevisionNumber</p> <p>The last number in the branch revision number. For example, if the branch revision number is 1.2, the version is 2.</p>

Related Reference

[Fields](#)

[File Fields](#)

[Change Request Fields](#)

[Requirement Fields](#)

[Task Fields](#)

[Folder Fields](#)

[Audit Fields](#)

Folder Fields

This section lists all the folder fields in alphabetical order.

Note: Client-calculated fields cannot be used in custom email notifications or Notification Agent. Reports can use any field name.

Field	Description
Branch On Change (Advanced)	<p>Values: No Yes</p> <p>Internal Identifier: BranchOnChange</p> <p>Indicates whether a folder will branch when it changes.</p> <p>The value is No if the folder's behavior is not set to Branch On Change. Reasons for this may be:</p> <ul style="list-style-type: none">- The folder is in the root or a reference view and the Branch On Change feature is disabled.- The folder is in a branching view but has already branched as a result of a change, which, in turn, results in the Branch On Change feature becoming disabled.- The folder is in a branching view, but its behavior currently does not permit it to branch on change. This means that modifications are checked into the parent view. <p>Note: If the value is No, the value of the Branch State explains the No.</p>
Branch State (Advanced)	<p>Values: Branched, Not Branched, Root</p> <p>Internal Identifier: BranchState</p> <p>Indicates whether a folder has branched in the child view, is still unbranched (and therefore is part of the parent view), or was created in the view in which it resides.</p> <p>The values Branched and Not Branched apply to folders in branching views. The value Root applies to folders created in the view in which the folder currently resides.</p> <p>If the view is a reference view, it reflects the state of the folder in the reference view's parent.</p>
Comment	<p>Values: text</p> <p>Internal Identifier: Comment</p> <p>The initial 2000 characters provided as the reason for changing a folder's properties or contents are stored in the Short Comment field. The Comment field stores those 2000 characters and any additional text. Changing a folder's properties causes the application to create a new revision. Note: To include a Link comment, the Comment field is the value to use in an HTML Report,</p>
CommentID (Advanced)	<p>Values: number</p> <p>Internal Identifier: CommentID</p> <p>The ID number assigned to revision comment. Displays -1 if no revision comment was supplied.</p>
Configuration Time	<p>Values: date/time</p> <p>Internal Identifier: ConfigurationTime</p>

	Indicates the time to which a folder is configured. If you configure a folder to a specific time, this field contains that time. If you configure a folder to a label or promotion state, this field shows either the time at which the label was created, or the time at which the label associated with the promotion state was created.
Created By	<p>Values: list of users, <None></p> <p>Internal Identifier: CreatedUserID</p> <p>The name of the user who created the first revision in the view. This is either the user who added the folder to the project, or the user who checked in the revision that branched.</p>
Created Time	<p>Values: date/time</p> <p>Internal Identifier: CreatedTime</p> <p>The time at which the first revision in the view was created.</p>
Creating Project	<p>Values:</p> <p>Internal Identifier: CreatingProject</p> <p>NEED CONTENT</p>
Deleted By	<p>Values: list of users, <None></p> <p>Internal Identifier: DeletedUserID</p> <p>The name of the user who deleted a folder. Because deleted folders do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: DeletedTime</p> <p>The time at which a folder was deleted. Because deleted folders do not appear in the list, this information is unavailable to users. Internal Use Only.</p>
Description	<p>Values: text</p> <p>Internal Identifier: Description</p> <p>The description provided for a folder at the time it was added to the view, including any later edits to it.</p>
Dot Notation	<p>Values: text</p> <p>Internal Identifier: DotNotation</p> <p>The branch revision number, for example, 1.2.1.0.</p>
Dot Notation ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: DotNotationID</p> <p>The ID assigned to a particular branch revision number. For example, if a folder was added to the current view (as opposed to inherited by the current view), its branch revision number is 1.x and its branch revision ID is 0. If a folder was branched in the current view, its branch revision ID is dependent on the revision number in the parent view and the number of IDs already assigned in the current view. For example, if a folder's revision number in the parent view is 1.7 at the time of the branch, and another folder with that same parent revision number was given the Branch Revision ID 6, this folder will also be given the Branch Revision ID 6.</p>
End Modified Time (Advanced)	<p>Values: date/time</p> <p>Internal Identifier: EndModifiedTime</p>

	<p>The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.</p>
Exclude Flags	<p>Values: <code>Inherit</code> and <code>Local Exclude List</code>, <code>Local Exclude List</code>, <code>No Exclude List</code></p> <p>Internal Identifier: <code>ExcludeFlags</code></p>
Exclude Spec	<p>The flag which specifies the types of file to be excluded from the folder.</p> <p>Values: text</p> <p>Internal Identifier: <code>ExcludeSpec</code></p> <p>The file specification (using the standard <code>*</code> and <code>?</code> wild cards), separated by commas, spaces or semicolons. To include a comma, space, or semicolon as part of the specification, enclose the specification in double quotes.</p>
Folder Path	<p>Values: text</p> <p>Internal Identifier: <code>Folder Path</code> (contains spaces)</p>
Is Action Overridden?	<p>The path to the folder. This is not the path to the working folder.</p> <p>Values: <code>No</code>, <code>Yes</code></p> <p>Internal Identifier: <code>Is Action Overridden?</code> (contains spaces)</p>
Local Path	<p>NEEDS CONTENT</p> <p>Values: text</p> <p>Internal Identifier: <code>Local Path</code> (contains spaces)</p>
Locked By	<p>The local path to the folder. This is the path to the working folder.</p> <p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>ExclusiveLocker</code></p>
Modified By	<p>The name of the user who has exclusively locked a folder.</p> <p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>ModifiedUserID</code></p>
Modified Time	<p>The name of the user who last modified a folder.</p> <p>Values: date/time</p> <p>Internal Identifier: <code>ModifiedTime</code></p> <p>The time at which a folder was last modified. The folder may have been checked in or had its properties changed. This has nothing to do with the working folder. Use Local Time Stamp for the time a working folder was last modified.</p>
Name	<p>Values: text</p> <p>Internal Identifier: <code>Name</code></p>
Non-Exclusive Lockers	<p>Displays the name of the folder.</p> <p>Values: text</p> <p>Internal Identifier: <code>NonExclusiveLockers</code></p>
Object ID	<p>The names of the users who have locked the folder non-exclusively.</p> <p>Values: number</p> <p>Internal Identifier: <code>ID</code></p>

	Each folder is assigned an object ID when it is added to a view. When it is branched in a child view, it is assigned another object ID. The original ID belongs to the folder in the parent view.
Parent Branch Revision (Advanced)	<p>Values: number</p> <p>Internal Identifier: <code>ParentRevision</code></p> <p>The last digit in the branch revision number before a folder branched. For example, if this number is <code>7</code>, the branch revision was <code>1.7</code> at the time the folder branched (becoming <code>1.7.1.0</code>, as seen in the folder's history). This number is <code>-1</code> if a folder was not inherited from the parent view.</p>
Parent ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: <code>ParentID</code></p> <p>The object ID of a folder in the parent view. The Parent ID is <code>-1</code> if this view has no parent view.</p>
Parent Revision (Advanced)	<p>Values:</p> <p>Internal Identifier: <code>PathRevision</code></p> <p>The revision number at which a folder branched. For example, if this number is <code>8</code>, this folder's revision number in the parent view was <code>8</code> at the time the folder branched. The history should show that revision <code>9</code> in the first revision in the current view. This number is <code>0</code> if this folder was not inherited from the parent view.</p>
Read Only (Advanced)	<p>Values: <code>No</code>, <code>Yes</code></p> <p>Internal Identifier: <code>ReadOnly</code></p> <p>Indicates whether the folder's configuration is read-only (as in a rollback configuration of a view) and/or its behavior does not allow it to branch on modification. Do not confuse a read-only configuration (an application issue) with a read-only folder (an operating system issue). A read-only folder cannot be edited and saved to disk. A folder whose configuration is read-only can be edited and saved to disk; it just cannot be checked in.</p>
Revision Flags (Advanced)	<p>Values: <code>0</code></p> <p>Internal Identifier: <code>RevisionFlags</code></p> <p>Internal use only.</p>
Root Object ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: <code>RootObjectID</code></p> <p>The object ID of the oldest ancestor of a folder. For example, if a folder was not inherited from a parent view, the root object ID is the same as its object ID. If it was inherited from a parent view, the root object ID is the Parent ID, or the parent's Parent ID.</p>
Share State	<p>Values: <code>DerivedShare</code>, <code>Not Shared</code>, <code>Root Share</code></p> <p>Internal Identifier: <code>ShareState</code></p> <p>Indicates whether this item is shared. <code>Not Shared</code> means that the item is not shared. <code>Root Share</code> means that the item is shared and this item is the original (or root) reference. <code>DerivedShare</code> means that the item is shared, but this item is not the original (or root) reference.</p>
Short Comment	<p>Values: text</p> <p>Internal Identifier: <code>ShortComment</code></p>

Status	<p>Stores the initial 2000 characters provided as the reason for changing a folder's properties or contents. Additional text is stored in the Comment field.</p> <p>Values: Current, Merge, Missing, Modified, Not In View, Out Of Date, Unknown</p> <p>Internal Identifier: Status</p> <p>Indicates the relationship between the copy of a folder in your working folder and the tip revision in the repository.</p>
Version (Advanced)	<p>Values: number</p> <p>Internal Identifier: RevisionNumber</p> <p>The last number in the branch revision number. For example, if the branch revision number is 1.3.1.2, the version is 2.</p>
View	<p>Values: list of views, <None></p> <p>Internal Identifier: ViewID</p> <p>The name of the view in which the item last branched. For example, if a folder is inherited from a parent view but is branched in a child view, the value of this field in the child view changes from the name of the parent view to the name of the child view for the revision that branched and subsequent revisions in the child view.</p>
***Working Folder	<p>Values: number</p> <p>Internal Identifier: WorkingFolder</p> <p>The name of the working folder.</p>

Related Reference

[Fields](#)
[File Fields](#)
[Change Request Fields](#)
[Requirement Fields](#)
[Task Fields](#)
[Topic Fields](#)
[Audit Fields](#)

Audit Fields

This section lists all the audit fields in alphabetical order.

Fields	Description
Class Name 1	<p>Values: text</p> <p>Internal Identifier: <code>Class Name 1</code> (contains spaces)</p> <p>The name of the class of items, such as Label, Promotion State, Folder, File, Change Request, Topic, or Task.</p>
Class Name 2	<p>Values: text</p> <p>Internal Identifier: <code>Class Name 2</code> (contains spaces)</p> <p>The name of the class of items, such as Folder, File, Change Request, Label, Topic, or Task.</p>
Class Name 3	<p>Values: text</p> <p>Internal Identifier: <code>Class Name 3</code> (contains spaces)</p> <p>The name of the class of items, such as Folder, File, Change Request, Label, Topic, or Task.</p>
Created By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>CreatedUserID</code></p> <p>Always empty because the audit entry is created by the system.</p>
Created Time	<p>Values: date/time</p> <p>Internal Identifier: <code>CreatedTime</code></p> <p>The time at which this entry was created.</p>
Deleted By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>DeletedUserID</code></p> <p>The name of the user who deleted an audit entry. Because deleted entries do not appear in the list, this information is unavailable to users.</p>
Deleted Time	<p>Values: date/time</p> <p>Internal Identifier: <code>DeletedTime</code></p> <p>The time at which an audit entry was deleted. Because deleted entries do not appear in the list, this information is unavailable to users.</p>
Event	<p>Values: <code>Added</code>, <code>Branched</code>, <code>Comment Edited</code>, <code>Created</code>, <code>Deleted</code>, <code>Edited</code>, <code>Item Overwritten</code>, <code>Label Attached</code>, <code>Label Created</code>, <code>Label Deleted</code>, <code>Label Detached</code>, <code>Label Frozen</code>, <code>Label Modified</code>, <code>Label Moved</code>, <code>Label Unfrozen</code>, <code>Lock Broken</code>, <code>Locked</code>, <code>Modified</code>, <code>Moved From</code>, <code>Moved To</code>, <code>Promotion Model Modified</code>, <code>Promotion State Modified</code>, <code>Shared</code>, <code>Unlocked</code>, <code>Vault Converted</code></p> <p>Internal Identifier: <code>EventID</code></p> <p>The name of the operation being recorded.</p>
Folder	<p>Values: text</p> <p>Internal Identifier: <code>Folder</code></p> <p>The name of the folder that stores the audit entry.</p>
Folder Path	<p>Values: text</p> <p>Internal Identifier: <code>Folder Path</code> (contains spaces)</p>

Folder VMID (Advanced)	<p>The path to the folder that stores the audit entry.</p> <p>Values: number</p> <p>Internal Identifier: <code>FolderVMID</code></p>
Item 1	<p>The ID assigned to the folder that stores the item.</p> <p>Values: text</p> <p>Internal Identifier: <code>Item 1</code> (contains spaces)</p> <p>Indicates what class <code>1</code> item received the audited operation. This can be the name of a file or task, the number of a change request or requirement, or the title of a topic.</p>
Item 1 Info	<p>Values: text</p> <p>Internal Identifier: <code>Info</code></p> <p>Provides the revision number in dot notation for the class <code>1</code> item, if it is revisionable. For example, a label can be a class <code>1</code> item and it does not have revisions.</p>
Item 2	<p>Values: text</p> <p>Internal Identifier: <code>Item 2</code> (contains spaces)</p> <p>Indicates what class <code>2</code> item received the audited operation. For example, if a label was attached to a file, the class <code>1</code> item is the label and the class <code>2</code> item is the file.</p>
Item 2 Info	<p>Values: text</p> <p>Internal Identifier: <code>Info2</code></p> <p>Provides the revision number in dot notation for the class <code>2</code> item, if it is revisionable. For example, a label can be a class <code>2</code> item and it does not have revisions.</p>
Item 3	<p>Values: text</p> <p>Internal Identifier: <code>Item 3</code> (contains spaces)</p> <p>Indicates what class <code>3</code> item received the audited operation. For example, if a label was moved from one revision to a file to another, the class <code>1</code> item is the label, the class <code>2</code> item is the revision of the file that was initially , and the class <code>3</code> item is the final revision of the file.</p>
Item 3 Info	<p>Values: text</p> <p>Internal Identifier: <code>Info3</code></p> <p>Provides the revision number in dot notation for the class <code>3</code> item, if it is revisionable. For example, a label can be a class <code>1</code> item and it does not have revisions.</p>
Modified By	<p>Values: list of users, <code><None></code></p> <p>Internal Identifier: <code>ModifiedUserID</code></p> <p>Does not apply to audit entries.</p>
Modified Time	<p>Values: date/time</p> <p>Internal Identifier: <code>ModifiedTime</code></p> <p>Does not apply to audit entries.</p>
Object ID	<p>Values: number</p> <p>Internal Identifier: <code>ID</code></p> <p>Each audit entry is assigned an object ID when it is added to a view.</p>

Project	<p>Values: list of projects in this server configuration, <None></p> <p>Internal Identifier: ProjectID</p> <p>The name of the project in which an audit entry was recorded.</p>
Target 1 Class ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: Target 1 Class ID (contains spaces)</p> <p>The ID number assigned to class 1 items or a -1 if there is no ID.</p>
Target 1 Object ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: Target 1 Object ID (contains spaces)</p> <p>The object ID for the class 1 item that received the audited operation or a -1 if there is no ID.</p>
Target 1 Revision Time	<p>Values: date/time</p> <p>Internal Identifier: Target 1 Revision Time (contains spaces)</p> <p>The time at which the last revision was made to the class 1 item that received the audit operation.</p>
Target 2 Class ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: Target 2 Class ID (contains spaces)</p> <p>The ID number assigned to class 2 items or a -1 if there is no ID.</p>
Target 2 Object ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: Target 2 Object ID (contains spaces)</p> <p>The object ID for the class 2 item that received the audited operation or a -1 if there is no ID.</p>
Target 2 Revision Time	<p>Values: number</p> <p>Internal Identifier: Target 2 Revision Time (contains spaces)</p> <p>The time at which the last revision was made to the class 2 item that received the audit operation.</p>
Target 3 Class ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: Target 3 Class ID (contains spaces)</p> <p>The ID number assigned to class 3 items or a -1 if there is no ID.</p>
Target 3 Object ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: Target 3 Object ID (contains spaces)</p> <p>The object ID for the class 3 item that received the audited operation or a -1 if there is no ID.</p>
Target 3 Revision Time	<p>Values: date/time</p> <p>Internal Identifier: Target 3 Revision Time (contains spaces)</p> <p>The time at which the last revision was made to the class 3 item that received the audit operation.</p>
Transaction ID (Advanced)	<p>Values: number</p> <p>Internal Identifier: TransactionID</p> <p>Uniquely identifies the database transaction that contained the update represented by the audit record. (A database transaction can contain multiple updates.) Note that audit records created before the database was upgraded to</p>

	a StarTeam release that records a Transaction ID will have a Transaction ID of -1 .
User	Values: list of users, <None> Internal Identifier: UserID
View	The name of the user who performed the recorded operation. Values: list of views, <None> Internal Identifier: ViewID The name of the view in which an audit entry was recorded.

Related Reference

[Fields](#)
[File Fields](#)
[Change Request Fields](#)
[Requirement Fields](#)
[Task Fields](#)
[Topic Fields](#)
[Folder Fields](#)

Relational Operators Used in Queries

The relational operators that you can use to define conditions in a query vary according to the type of field:

- ◆ Text fields.
- ◆ Boolean, enumerated type, and numeric fields.
- ◆ Date/time fields.

Relational Operators Used on Text Fields

The relational operators that can be used on text fields are:

Equals
Is Not
Contains (ignore case)
Contains (match case)
Starts with (ignore case)
Starts with (match case)
Ends with (ignore case)
Ends with (match case)

Relational Operators Used on Boolean, Enumerated Type, and Numeric fields

The relational operators that can be used on Boolean, enumerated type, and numeric fields are:

Less Than
Same or Less
Equals
Same or Greater
Greater Than
Is Not

Relational Operators Used on Date/Time Fields

The relational operators that can be used on date/time fields are listed below.

Relational operators that compare both the date and the time parts of date/time fields:

Before
On or Before
On
On or After
After
Not On

Relational operators that compare only the date part of date/time fields:

Before Date

On or Before Date

On Date

On or After Date

After Date

Relational operators that match all dates starting with the date that was the specified number of days or weeks ago:

Last (n) Days

Last (n) Weeks

Relational operators that match all the dates prior to and including the date that was the specified number of days or weeks ago:

Older Than (n) Days

Older Than (n) Weeks

Note: In date fields, StarTeam treats blanks as zeroes. That means that “no date” is less than any specific date. For example, if you write a query that searches for change requests that were closed prior to some specific date, all the change requests with no date in the **Closed On** field are included in the results—even though they have not been closed yet. It is easy to eliminate the change requests that contain blanks in the **Closed On** field from such a query. You simply AND the condition that searches for change requests closed on or before a specific date with another condition that searches for change requests closed after the date zero.

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

File Status

This section describes the status of files and the effect of file status on check-in and check-out operations.

In This Section

[File Status Information](#)

Describes the various file status states.

[Effects of Status on Check-ins and Check-outs](#)

Table summarizing the relationship between file status and the check-in and check-out operations for the application.

File Status Information

The operations you can perform on a particular file depend upon the status of the file. A successful check-in or check-out operation updates the status of your working file to Current, unless you check out a historical revision or delete the file. This list includes all status classes and shows their effects on check-ins and check-outs.

Status	Description
Current	<p>The tip revision of this file is in your working folder.</p> <p>Check-ins: Does not affect the file revision in the repository.</p> <p>Check-outs: Does not affect the file revision on your workstation.</p>
Missing	<p>The file is not in your working folder.</p> <p>Check-ins: Not applicable. If a file has a Missing status, it is not in your working folder, so it cannot be checked in.</p> <p>Check-outs: Copies the file to your working folder. If a file has a Missing status, you will be asked if you want to check it out when you open it. You can also check it out manually.</p>
Merge	<p>The file in your working folder has been modified, but is not based on the tip (latest) revision of this file. This status usually occurs when you and another person have both been working on a file, but the other person has checked it in before you.</p> <p>Check-ins: Unless you force a check-in, File Compare/Merge starts. If the file is a text file, File Compare/Merge can reconcile the differences between it and the tip revision. If the file is binary, you may be able to merge the working version and the tip revision by using the application in which the file was created.</p> <p>Check-outs: Not allowed, unless you force the check-out. The file changes you have made in your working folder will be lost.</p>
Modified	<p>Your working file has been altered and is based on the tip revision of this file.</p> <p>Check-ins: Unless someone else has the file locked, you can check in the file.</p> <p>Check-outs: Not allowed unless you force the check-out. Your changes will be lost if you check out the file, because it overwrites the file in your working folder.</p>
Out of Date	<p>The file in your working folder is an outdated revision of the file. If you need the current revision, you should check it out.</p> <p>Check-ins: Not allowed, unless you force the check-in. Checking in an Out Of Date file overwrites the changes made to the tip revision of the file with your older file revision.</p> <p>Checkouts: Checking out an Out Of Date file makes the file in your working folder Current.</p>
Not in view	<p>The file is in your working folder, but not in the project view.</p> <p>Check-ins: Not applicable. Files with the Not in View status cannot be checked in. However, you can add them to the project with the Add Files command.</p> <p>Check-outs: Not applicable. Files with the Not in View status are not in the project view, so they cannot be checked out.</p>
Unknown	<p>The file in the working folder has the same name as a file in the view, but the file in the view has not been checked out from the repository. You may have copied it from another location.</p> <p>Check-ins: Not allowed, unless you force the operation. If the file status is Unknown, the consequences of this action are also unknown, except that the file in your working folder becomes the tip revision in the repository. Before forcing a check-in, try to identify the file by using Update Status with an MD5 checksum instead of a time stamp (see File Options). If you still cannot identify the file, compare the file in your working folder to the current tip revision.</p> <p>Check-outs: Allowed, if you merge the file in your working folder with the tip revision. However, because the first revision acts as the ancestor file for the merge, many parts of the files may be in conflict. Instead of merging, you may prefer to force a check-out or check-in. If the file status is Unknown, the consequences of forcing a check-out are also unknown, except that this action also overwrites the file in your working directory. Before forcing a</p>

check-out, try to identify the file by using Update Status with an MD5 checksum. If you still cannot identify the file, compare the file in your working folder to the current tip revision.

Related Concepts

[Achieving Consistent Check-ins and Check-outs](#)

Related Procedures

[Checking In Files](#)

[Checking Out Files](#)

Related Reference

[Effects of Status on Check-ins and Check-outs](#)

[File Options \(Personal Options Dialog Box\)](#)

Effects of Status on Check-ins and Check-outs

During the file check-out process, the application copies a file revision from the repository to a working folder. Checking in a file places a new revision in the repository. In many cases, the status of a file affects the check-in or check-out process.

Status	Check-in	Check-out
Current	No considerations.	No considerations.
Merge	<p>Starts File Compare/Merge unless you force the check-in.</p> <p>The Merge status means that someone else has checked in this file since your last check-out. You do not have their changes in your working file and someone's changes will</p>	<p>Your changes will be lost if you check out this file.</p> <p>revision and your working file using the application in which this file was created, for example, Word for Windows. If the file is a text file, try a check-in operation.</p>
Missing	<p>Not applicable.</p> <p>If a file has the Missing status, it is not in your working folder so there is nothing to check in.</p>	<p>No considerations.</p> <p>If a file has the Missing status, you are asked if you want to check it out when you open it. You can check it out manually, too.</p>
Modified	<p>No considerations.</p> <p>Unless someone else has the file locked, you can check in the file.</p>	<p>No considerations.</p> <p>Unless someone else has the file locked, you can check out the file.</p>
Not in View	<p>Not applicable.</p> <p>A file with the Not in View status cannot be checked in. You can add it to the project with the Add Files command.</p>	<p>Not applicable.</p> <p>A file with the Not in View status is not in the repository, so there is nothing to check out.</p>
Out of Date	<p>Not allowed unless you force the check-in.</p> <p>Checking in an Out Of Date file means that the tip revision no longer has the changes made to the file since the time your working copy became Out Of Date.</p>	<p>No considerations.</p> <p>Checking out an Out Of Date file makes your working file Current.</p>
Unknown	<p>Not allowed unless you force the check-in.</p> <p>If the file's status is Unknown, the consequences of this action are also unknown. Your working file becomes the tip revision in the repository. Use Update Status with an MD5 checksum to see if the file can be identified.</p> <p>You might want to compare your working file to the tip revision if this is not successful.</p>	<p>Allowed if you merge the file with the tip revision. However, because the very first revision is used as the ancestor file for this merge, it is likely that many, many things appear to have changed or be in conflict. You may prefer to force a check-out (or force a check-in).</p> <p>If the file's status is Unknown, the consequences of this action are also unknown. Your working file is overwritten by the tip revision in the repository. Use Update Status with an MD5 checksum to see if the file can be identified.</p> <p>You might want to compare your working file to the tip revision if this is not successful.</p>

Related Concepts

[Achieving Consistent Check-ins and Check-outs](#)

Related Procedures

[Checking In Files](#)

[Checking Out Files](#)

Related Reference

[File Status Information](#)

[File Options \(Personal Options Dialog Box\)](#)

Filters Reference Topics

This section contains reference topics related about the default filters StarTeam provides for each component.

In This Section

[File Filters](#)

Describes the default filters provided for the File component.

[Change Request Filters](#)

Describes the default filters provided for the Change Request component.

[Requirement Filters](#)

Describes the default filters provided for the Requirement component.

[Folder Filters](#)

Describes the default filters provided for the Folder component.

[Task Filters](#)

Describes the default filters provided for the Task component.

[Topic Filters](#)

Describes the default filters provided for the Topic component.

[Audit Filters](#)

Describes the default filters provided for the Audit component.

[View Compare/Merge Session Filters](#)

File Filters

File ▶ Filters ▶ Filters

Filtering allows you to limit the types and numbers of files that appear in the upper pane. The list of filters depends on your selection from the folder hierarchy in the left pane and whether the **All Descendants** button is selected from the toolbar or **File** menu.

StarTeam ships with a set of default filters for the **File** component which are listed below. You can customize these filters or create new ones.

Tip: You can also view and apply the defined file filters using the **Filter** drop-down list box on the toolbar

Item	Description
<All Files By Status>	Groups the files that have the same status: Current , Missing , Modified , Merge , Out Of Date , Not In View , and Unknown .
<Flagged Items>	Files that have been flagged for some special reason. For example, you may want to use flags to remind yourself to follow up on a customer request. Flags are set, viewed, and removed by the user who created them.
All Non-Excluded Files	All non-excluded files that exist either in application folders or their working folders.
Files In View	Files in the working folder that exist in the current project view.
Files Not In View	Files in the working folder that do not exist in the current project view. Unless you add them to the project, their names will never appear on the same list as the files that are in your project.
Files to Check In	All files in the view that need to be checked in. The statuses are Modified , Merge , or Not In View .
Files to Check Out	All files in the view that need to be checked out. The statuses are Out Of Date , Missing , or Merge .

Note: StarTeam lists the files that need to be merged when you apply either the **Files To Check In** or **Files To Check Out** filter.

Related Concepts

[Filters](#)

Related Procedures

[Filtering Data](#)

Change Request Filters

Change Request ► Filters ► Filters

Use the **Filters** dialog box to limit the kinds and quantity of change requests that appear in the change request list. The list of filters depends on your selection from the folder hierarchy in the left pane and whether the **All Descendants** button is selected from the toolbar or **Change Request** menu.

StarTeam ships with a set of default filters for the **Change Request** component which are listed below. You can customize these filters or create new ones.

Tip: You can also view and apply the defined change request filters using the **Filter** drop-down list box on the toolbar

Item	Description
<Show All>	Displays all change requests.
By Status and Responsibility	Groups change requests based on their statuses and the users who are currently responsible for processing the requests.
Not a Priority	Displays only the change requests that are not a priority.
Priority	Displays only the change requests that are a priority.
Show Unread Changes	Displays only the change requests that you have not read (or not read since they were modified).
Status = Closed	Displays only the change requests that are closed.
Status = Deferred	Displays only the change requests that are postponed.
Status = Open	Displays only the change requests that are open and in progress.
Status = Resolved	Displays all the change requests that have one of the following statuses: As Designed, Cannot Reproduce, Documented, Fixed, or Is Duplicate.
Status = Verified	Displays all the change requests that have one of the following statuses: Verified As Designed, Verified Cannot Reproduce, Verified Documented, Verified Fixed, or Verified Is Duplicate.
Type = Defect	Displays only the change requests that have the type Defect.
Type = Suggestion	Displays only the change requests that have the type Suggestion.

Related Concepts

[Filters](#)

Related Procedures

[Filtering Data](#)

Requirement Filters

[Requirement](#) ▶ [Filters](#) ▶ [Filters](#)

Filtering allows you to limit the types and numbers of requirements that appear in the upper pane. The list of filters depends on your selection from the folder hierarchy in the left pane and whether the **All Descendants** button is selected from the toolbar or [Requirement](#) menu.

StarTeam ships with a set of default filters for the **Requirement** component which are listed below. You can customize these filters or create new ones.

Tip: You can also view and apply the defined requirement filters using the **Filter** drop-down list box on the toolbar

Item	Description
<Show All>	Displays all the requirements.
Flagged Items	Lists only requirements that have been flagged.
Grouped by Creator	Displays groups of requirements, one group for each user who has created requirements.
Grouped by Status	Displays groups of requirements, one group for each existing status.
I Am Responsible	Displays only the requirements for which you are responsible.

Related Concepts

[Requirements](#)

Related Procedures

[Using Requirements](#)

Folder Filters

[Folder](#) ▶ [Filters](#) ▶ [Filters](#)

Filtering allows you to limit the types and numbers of folders that appear in the upper pane. The list of filters depends on your selection from the folder hierarchy in the left pane and whether the **All Descendants** button is selected from the toolbar or [Folder](#) menu.

StarTeam ships with a set of default filters for the **Folder** component which are listed below. You can customize these filters or create new ones.

Tip: You can also view and apply the defined file filters using the **Filter** drop-down list box on the toolbar

Item	Description
<Show All>	Displays all folders (the default).
Folders Not In View	Displays only folders that are not in the current view.

Related Concepts

[Folders](#)

Related Procedures

[Working with Folders and Items](#)

Task Filters

Task ▶ Filters ▶ Filters

Filtering allows you to limit the types and numbers of tasks that appear in the upper pane. The list of filters depends on your selection from the folder hierarchy in the left pane and whether the **All Descendants** button is selected from the toolbar or **Task** menu.

StarTeam ships with a set of default filters for the **Task** component which are listed below. You can customize these filters or create new ones.

Examples of custom task filters that you might create include:

- ◆ `Responsibility Equals <user name>`, which identifies only the tasks for which a specific person is responsible.
- ◆ `Percent Complete Less Than 100`, which identifies unfinished tasks.

Tip: You can also view and apply the defined task filters using the **Filter** drop-down list box on the toolbar

Item	Description
<Show All>	Lists all tasks.

Related Concepts

[Tasks](#)

Related Procedures

[Using Tasks](#)

Topic Filters

Topic ▶ Filters ▶ Filters

Filtering allows you to limit the types and numbers of topics that appear in the upper pane. The list of filters depends on your selection from the folder hierarchy in the left pane and whether the **All Descendants** button is selected from the toolbar or **Topic** menu.

StarTeam ships with a set of default filters for the **Topic** component which are listed below. You can customize these filters or create new ones.

Tip: You can also view and apply the defined topic filters using the **Filter** drop-down list box on the toolbar

Item	Description
<I Am Recipient>	Identifies all the topics that name you as a recipient.
By Creator	Groups the topics by their original authors.
Show Active	Identifies all topics and responses that have Active status.
Show All	Identifies all topics (the default).

Related Concepts

[Topics](#)

Related Procedures

[Using Topics](#)

Audit Filters

Audit ► Filters ► Filters

Filtering allows you to specify what fields are displayed in the audit entries that appear in the upper pane and how those fields are grouped and sorted. The list of audit records depends on your selection from the folder hierarchy in the left pane and whether the **All Descendants** button is selected from the toolbar or **Audit** menu.

StarTeam ships with a set of default filters for the **Audit** component which are listed below. Use the **Filter** drop-down list box on the toolbar to view and apply the defined audit filters.

Note: You can customize these filters or create new ones. See **Creating Filters** under **Related Procedures** below.

Item	Description
<By Class and Event>	Displays audit entries sorted by their value in the Class Name 1 field (type of item) and Event (type of action) field.
By Transaction and Event	Groups audit log entries by descending Transaction ID and then by Event type. This filter provides a reverse-chronological list of updates in the view by transaction.
Events	Groups audit log entries by Event type, then by Target 1 Class ID , and then by Created Time .
Show All	Displays all entries (the default).

Note: You can limit the number of audit log entries displayed by creating a query that selects audit log entries by specific property values. See **Queries** under **Related Concepts** below.

Related Concepts

[Queries](#)

Related Procedures

[Creating Filters](#)

[Viewing the Audit Log](#)

View Compare/Merge Session Filters

Filtering allows you to limit the types and numbers of folders that appear in the upper pane. The list of available filters depends on the perspective you are using in the VCM Session. StarTeam provides a set of predefined filters for View Compare/Merge which are listed below. The predefined filters are intended as starting points for you to create your own custom filters. Use the **Filter** drop-down list box on the toolbar to view and apply predefined file filters.

You can create additional filters for the **Merge Perspective** and the **Test Perspective**. To access the **Filters** dialog box, right-click a column header in the upper pane of the **Merge Perspective** or **Test Perspective** and choose **Filters**. The **Compare Perspective** has two predefined filter drop-down lists, but you cannot create any additional filters for that perspective.

For more details on using filters, see "Filtering Items in a View Compare/Merge Session" in the links below.

Below are the View Compare/Merge session predefined filters:

Item	Description
<Show Type>	Displays only items of the selected type in the Compare Perspective .
<Show Items With Differences>	Displays items from both the source and target views in the Compare Perspective that have differences.
<Show Unresolved Items>	Displays only items in the Compare Perspective with a merge status of Unresolved .
<Show All Items>	Displays all items of the selected type in the Compare Perspective .
Show Items: [action]	Displays only items in the Compare Perspective with the selected merge action.
<VCM: Items By Merge Status>	Displays items in the upper pane of the Merge Perspective sorted by merge status. This filter name is dynamic, depending on which component tab is selected. You can display items by merge status for files, change requests, and folders.
<VCM: Merge Properties>	Displays the items in the upper pane of the Merge Perspective with the primary descriptor in column one, and all the remaining VCM properties in the following columns.
Merge Action	Sorts items in the upper pane of the Merge Perspective and Test Perspective the by merge action.
<Show Items With Updates>	Displays only items from the source view in the Compare Perspective that will result in updates to the target.

Note: In the **Test Perspective**, you also see the filters for files when the **Files** tab is selected, for folders when the **Folders** tab is selected, and so on.

Related Concepts

[Filters](#)

Related Procedures

[Filtering Items in a View Compare/Merge Session](#)

[Creating Filters](#)

[Applying Predefined Filters](#)

[Copying Filters](#)

Links

This section contains UI overview topics related to Links.

In This Section

[Link Tab](#)

Describes the items on the Link tab.

Link Tab

The **Link** tab displays all the links associated with an item selected in the upper pane. Radio buttons also let you specify whether to show all types of links, or just the enhanced process links.

Each link has two ends: a source and a target. The source is the item selected from the upper pane or from the folder hierarchy (if you are viewing the Link tab of the **Folder Properties** dialog). Each row on the **Link** tab defines a link that has the source item or folder as one of its ends. Columns in the link list identify the other end of the link for you.

The table below explains the columns provided in the Link tab. These columns also appear on the **Link** tab of the **Folder Properties** dialog. (The **Folder Properties** dialog has no **Link** tab if you do not have the access rights that allow you to view links.)

Column	Description
Created By	The name of the user who created the link.
Created On	The date/time the link was created.
View	The assumption is that you want to locate items in the current view whenever possible, so the view field contains one of the following: <ul style="list-style-type: none">– The name of the current view if the link was created in the current view, or if the link was created in another view but a shared copy of the item (to which the current item is linked) exists in the current view.– The name of the view containing the link. If the view is a reference view, this is the name of the parent view.
Folder	The name of the folder in which the folder or item in the link list resides.
Item Type	Identifies the type of item the target end of the link is attached to. This item is the one listed in the link list. The values that can appear in this column are Folder, File, Change Request, Requirement, Task, and Topic.
Item	Identifies the item that the target end of the link is attached to by its folder name, filename, change request number, task number, and so on.
Item Details	Describes the item, using a folder description, file description, change request synopsis, requirement name, task name, or topic title.
Item Version	When the ends of a link are pinned to specific revisions of the folders or items, those revisions appear in the Item Version and the Selection Version columns of the link list (if that revision is in the current view).
Selection Version	The Item Version displays the version number of the target end of the link if that revision is in the current view. The Selection Version displays the revision number the source end of the link. The source is either the folder selected from the folder hierarchy (when viewing a link for a folder) or the item selected from the upper pane (when viewing the link for an item). When no revision number is in the column, that end of the link is floating rather than pinned.
Comment	From the Link Properties dialog, you can enter a comment about this particular link. That comment appears in this column.
Folder Path	Shows folder path information only when the linked item is in the same view. Otherwise, displays the following message: "Unavailable. Item in another view."

Related Procedures

[Linking Items Internally or Externally](#)

[Enabling the use of Enhanced Process Links](#)

[Reviewing Linked Change Requests](#)

[Selecting Linked Files](#)

[Customizing Link Properties](#)

[Linking Files to Process Items](#)

Queries

This section contains reference topics related to queries.

In This Section

[Predefined Queries](#)

A list of the default queries that are predefined for each component.

[Relational Operators Used in Queries](#)

Describes the relational operators that you can use to define conditions in a query.

[New Query Options](#)

Describes the options in the New Query dialog box.

[Queries Options](#)

Describes the options in the Queries dialog box.

Predefined Queries

[Component](#) ▶ [Filters](#) ▶ [Queries](#)

Displays the **Queries** dialog box. Initially, the dialog box lists default queries that are predefined for the current component. You can apply a default query, edit a default query, create a new query, or delete a default query.

Note: Default queries that do not appear in your list might have been deleted after installation.

Component	Predefined Queries
File	Files to Check In, Files to Check Out, Flagged Items
Change Request	Flagged Items, Not a Priority, Priority, Status=Closed, Status=Open, Status=Resolved, Status=Verified, Type=Defect, Type=Suggestion, Unread Changes
Requirement	Flagged Items, I Am Responsible
Task	Flagged Items
Topic	Flagged Items, I Am Recipient, Show Active
Folder	Folders Not In View
Audit	None

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

Relational Operators Used in Queries

The relational operators that you can use to define conditions in a query vary according to the type of field:

- ◆ Text fields.
- ◆ Boolean, enumerated type, and numeric fields.
- ◆ Date/time fields.

Relational Operators Used on Text Fields

The relational operators that can be used on text fields are:

Equals
Is Not
Contains (ignore case)
Contains (match case)
Starts with (ignore case)
Starts with (match case)
Ends with (ignore case)
Ends with (match case)

Relational Operators Used on Boolean, Enumerated Type, and Numeric fields

The relational operators that can be used on Boolean, enumerated type, and numeric fields are:

Less Than
Same or Less
Equals
Same or Greater
Greater Than
Is Not

Relational Operators Used on Date/Time Fields

The relational operators that can be used on date/time fields are listed below.

Relational operators that compare both the date and the time parts of date/time fields:

Before
On or Before
On
On or After
After
Not On

Relational operators that compare only the date part of date/time fields:

Before Date

On or Before Date

On Date

On or After Date

After Date

Relational operators that match all dates starting with the date that was the specified number of days or weeks ago:

Last (n) Days

Last (n) Weeks

Relational operators that match all the dates prior to and including the date that was the specified number of days or weeks ago:

Older Than (n) Days

Older Than (n) Weeks

Note: In date fields, StarTeam treats blanks as zeroes. That means that “no date” is less than any specific date. For example, if you write a query that searches for change requests that were closed prior to some specific date, all the change requests with no date in the **Closed On** field are included in the results—even though they have not been closed yet. It is easy to eliminate the change requests that contain blanks in the **Closed On** field from such a query. You simply AND the condition that searches for change requests closed on or before a specific date with another condition that searches for change requests closed after the date zero.

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

New Query Options

Component ► Filters ► Queries

Click **New** to define a new query in the **New Query** dialog box.

Note: You can also access the **Queries** dialog box from the context menu by right-clicking on a column title in any component.

Item	Description																		
Query	Displays the query definition parameters. <table> <tr> <td>Name</td><td>Enter a unique name for the new query.</td></tr> <tr> <td>Public check box</td><td>Check to give the query public status. Public queries can be used by anyone with the appropriate access rights, while private queries are available only to your user ID. Once a query has been saved with a specific status, its status cannot be changed. However, you can copy a query and change the state of the new query.</td></tr> <tr> <td>Query list box</td><td>Displays the query definition. By default, the AND condition appears as a starting logical condition.</td></tr> </table>	Name	Enter a unique name for the new query.	Public check box	Check to give the query public status. Public queries can be used by anyone with the appropriate access rights, while private queries are available only to your user ID. Once a query has been saved with a specific status, its status cannot be changed. However, you can copy a query and change the state of the new query.	Query list box	Displays the query definition. By default, the AND condition appears as a starting logical condition.												
Name	Enter a unique name for the new query.																		
Public check box	Check to give the query public status. Public queries can be used by anyone with the appropriate access rights, while private queries are available only to your user ID. Once a query has been saved with a specific status, its status cannot be changed. However, you can copy a query and change the state of the new query.																		
Query list box	Displays the query definition. By default, the AND condition appears as a starting logical condition.																		
Logical node	Use to choose the logical operator for each condition. <table> <tr> <td>AND button</td><td>Click to add an AND condition to the query.</td></tr> <tr> <td>OR button</td><td>Click to add an OR condition to the query.</td></tr> <tr> <td>NOT button</td><td>Click to add a NOT condition to the query.</td></tr> <tr> <td>AND->OR->NOT button</td><td>Click to toggle the default AND condition to an OR or NOT condition.</td></tr> <tr> <td>Remove button</td><td>Click to delete the currently selected condition. You must confirm the deletion.</td></tr> </table>	AND button	Click to add an AND condition to the query.	OR button	Click to add an OR condition to the query.	NOT button	Click to add a NOT condition to the query.	AND->OR->NOT button	Click to toggle the default AND condition to an OR or NOT condition.	Remove button	Click to delete the currently selected condition. You must confirm the deletion.								
AND button	Click to add an AND condition to the query.																		
OR button	Click to add an OR condition to the query.																		
NOT button	Click to add a NOT condition to the query.																		
AND->OR->NOT button	Click to toggle the default AND condition to an OR or NOT condition.																		
Remove button	Click to delete the currently selected condition. You must confirm the deletion.																		
Condition Node	Use to define the conditions for the logical operator. <table> <tr> <td>Field</td><td>Lists all the fields available for this component.</td></tr> <tr> <td>Operator</td><td>Lists all of the operators that can be specified for the selected field.</td></tr> <tr> <td>Value</td><td>Use to specify a value for this field and condition.</td></tr> <tr> <td>Show advanced fields check box</td><td>Select to display all possible fields including the advanced fields.</td></tr> <tr> <td>Show deleted users check box (optional)</td><td>For components with user fields only (such as the Change Requests component), select to show deleted users in the query results.</td></tr> <tr> <td>Alphabetical check box (optional)</td><td>For enumerated fields only (fields that have specified values), select to alphabetize the query results rather than list them in the order in which they appear in the enumeration list.</td></tr> <tr> <td>Add button</td><td>Click to add the condition to the query definition.</td></tr> <tr> <td>Modify button</td><td>Click to modify the selected query condition.</td></tr> <tr> <td>Delete button</td><td>Click to delete the selected query condition.</td></tr> </table>	Field	Lists all the fields available for this component.	Operator	Lists all of the operators that can be specified for the selected field.	Value	Use to specify a value for this field and condition.	Show advanced fields check box	Select to display all possible fields including the advanced fields.	Show deleted users check box (optional)	For components with user fields only (such as the Change Requests component), select to show deleted users in the query results.	Alphabetical check box (optional)	For enumerated fields only (fields that have specified values), select to alphabetize the query results rather than list them in the order in which they appear in the enumeration list.	Add button	Click to add the condition to the query definition.	Modify button	Click to modify the selected query condition.	Delete button	Click to delete the selected query condition.
Field	Lists all the fields available for this component.																		
Operator	Lists all of the operators that can be specified for the selected field.																		
Value	Use to specify a value for this field and condition.																		
Show advanced fields check box	Select to display all possible fields including the advanced fields.																		
Show deleted users check box (optional)	For components with user fields only (such as the Change Requests component), select to show deleted users in the query results.																		
Alphabetical check box (optional)	For enumerated fields only (fields that have specified values), select to alphabetize the query results rather than list them in the order in which they appear in the enumeration list.																		
Add button	Click to add the condition to the query definition.																		
Modify button	Click to modify the selected query condition.																		
Delete button	Click to delete the selected query condition.																		
View as Text button	Displays the current query definition in a text box.																		
Save button	Saves the new query.																		
Cancel button	Cancels the query definition.																		

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

Queries Options

[Component](#) ▶ [Filters](#) ▶ [Queries](#)

Use the **Queries** dialog box to view and apply the currently defined queries.

Note: You can also access the **Queries** dialog box from the context menu by right-clicking on a column title in any component.

Item	Description
Queries list box	Lists all existing queries for this component. Note that the multi-person icon left of the query name indicates a public query; a single-person icon indicates a private query.
Close button	Closes the Queries dialog box.
Select button	Applies the selected query to the data listed in the selected component.
Clear Query button	Clears the current query and displays all data.
New button	Opens the New Query dialog box for you to define a new query.
Edit button	Opens the selected query definition in the Edit Query dialog box for you to edit.
Copy button	Opens the Copy Query dialog box for you to enter a name for your new query.
Delete button	Deletes the selected query definition.
Access Rights button	Allows you to assign access rights to a query.

Related Concepts

[Queries](#)

Related Procedures

[Querying Data](#)

Reports

This section contains reference topics related to reports and report templates.

In This Section

[Available Audit Reports](#)

Describes the default reports provided for the Audit component.

[Available Change Request Reports](#)

Describes the default reports provided for the Change Request component.

[Available File Reports](#)

Describes the default reports provided for the File component.

[Available Folder Reports](#)

Describes the default reports provided for the Folder component.

[Available Requirement Reports](#)

Describes the default reports provided for the Requirement component.

[Available Task Reports](#)

Describes the default reports provided for the Task component.

[Available Topic Reports](#)

Describes the default reports provided for the Topic component.

[Report Templates](#)

Lists the component reports and the template files used to generate them.

Available Audit Reports

Audit ► Reports

You can view and print a number of different reports for the Audit component.

Name of Report	Description
Default	Lists information located in the Detail pane, using one line for each field.
Grouping Summary	Indicates the number of audit entries in each group plus the total number of audit entries.
Summary	Lists each selected audit entry (or all of them when none are selected). The report gives the Event , Created Time , User , Class 1 , Item 1 , Class 2 , Item 2 , Class 3 , and Item 3 fields and the total number of selected audit entries.

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports and Exporting Data](#)

Related Reference

[Available File Reports](#)

[Available Change Request Reports](#)

[Available Requirement Reports](#)

[Available Task Reports](#)

[Available Topic Reports](#)

[Available Folder Reports](#)

Available Change Request Reports

Change Request ► Reports

You can view and print a number of different reports for the Change Request component.

Name of Report	Description
Default	Lists information located in the Detail pane, using one line for each field.
Detail	Lists specific change request fields: Modified Time, Responsibility, Entered On, Entered By, Last Build Tested, Status, Severity, Addressed In, Addressed By, Priority, Test Command, Type, Synopsis, Description, Work Around, and Fix.
Grouping Summary	Indicates the number of change requests in each group plus the total number of change requests.
History	Lists the change request fields found in a Detail report followed by information about each revision of the change request: the revision number, its date and time, author, view, comment, and branch revision.
Links	Lists the selected change requests and the items linked to them.
Summary	Lists each selected change request (or all of them when none are selected). The report gives the Modified Time, Responsibility, Entered On, Entered By, Priority, Type, Status, Severity, and Synopsis fields and the total number of selected change requests.

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports and Exporting Data](#)

Related Reference

[Available File Reports](#)

[Available Requirement Reports](#)

[Available Task Reports](#)

[Available Topic Reports](#)

[Available Folder Reports](#)

[Available Audit Reports](#)

Available File Reports

File ► Reports

You can view and print a number of different reports for the File component.

Name of Report	Description
Default	Lists information located in the Detail pane, using one line for each field.
Detail	Lists files and their revision histories.
Detail with Description	Lists files, their descriptions and revision histories.
Grouping Summary	Indicates the number of files in each group plus the total number of files.
Links	Lists the selected file names and the items linked to them.
Summary	Lists each selected file (or all of them when none are selected). The report gives the Name , Status , Locked By , and Revision fields and the total number of selected files.
Summary with Description	Lists each selected file (or all of them when none are selected). The report gives the Name , Status , Locked By , Revision , and Description fields and the total number of selected files.

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports and Exporting Data](#)

Related Reference

[Available Change Request Reports](#)
[Available Requirement Reports](#)
[Available Task Reports](#)
[Available Topic Reports](#)
[Available Folder Reports](#)
[Available Audit Reports](#)

Available Folder Reports

Folder ► Reports

You can view and print a number of different reports for the Folder component.

Name of Report	Description
Default	Lists information located in the Detail pane, using one line for each field.
Detail	Lists folders and their revision histories.
Detail with Description	Lists folders, their descriptions and revision histories.
Grouping Summary	Indicates the number of folders in each group plus the total number of folders.
Links	Lists the selected folders names and the items linked to them.
Summary	Lists each selected file (or all of them when none are selected). The report gives the File , Status , Locked By , and Revision fields and the total number of selected folders.
Summary with Description	Lists each selected folder (or all of them when none are selected). The report gives the File , Status , Locked By , Revision , and Description fields and the total number of selected folders.

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports and Exporting Data](#)

Related Reference

[Available File Reports](#)

[Available Change Request Reports](#)

[Available Requirement Reports](#)

[Available Task Reports](#)

[Available Topic Reports](#)

[Available Audit Reports](#)

Available Requirement Reports

Requirement ► Reports

You can view and print a number of different reports for the Requirement component.

Name of Report	Description
Default	Lists information found in the Detail pane (when in list format), using one line for each field. You can create this report while the pane is in either tree or list format.
Detail	Lists specific requirement fields: Modified Time , Name , Type , Status , Owner , Priority , and Description .
Grouping Summary	Indicates the number of requirements in each group plus the total number of requirements.
History	Lists the change request fields found in a Detail report followed by information about each revision of the change request: the revision number, its date and time, author, view, comment, and branch revision.
Links	Lists the selected requirements and any items linked to them.
Summary	Lists each selected requirement (or all of them when none are selected). The report gives the Title , Created By , and Created Time fields and the total number of selected requirements.

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports and Exporting Data](#)

Related Reference

[Available File Reports](#)

[Available Change Request Reports](#)

[Available Task Reports](#)

[Available Topic Reports](#)

[Available Folder Reports](#)

[Available Audit Reports](#)

Available Task Reports

Task ► Reports

You can view and print a number of different reports for the Task component.

Name of Report	Description
Detail	Lists specific Task fields: Modified Time, Responsibility, Created Time, Created By, Name, Status, Priority, Milestone, Duration, Percent Complete, Needs Attention, Attention Notes, Planned Start, Planned Finish, Planned Work, Actual Start, Actual Finish, Actual Work, and Notes.
Grouping Summary	Indicates the number of tasks in each group plus the total number of tasks.
Links	Lists the selected tasks and any items linked to them.
Summary	Lists each selected task (or all of them when none are selected). The report gives the Task Name, Created By, and Created Time fields and the total number of selected tasks.

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports and Exporting Data](#)

Related Reference

[Available File Reports](#)

[Available Change Request Reports](#)

[Available Requirement Reports](#)

[Available Topic Reports](#)

[Available Folder Reports](#)

[Available Audit Reports](#)

Available Topic Reports

Topic ► Reports

You can view and print a number of different reports for the Topic component.

Name of Report	Description
Default	Lists information located in the Detail pane, using one line for each field. You can create this report while the pane is in either tree or list format.
Detail	Lists specific Task fields: Created By , Status , Created Time , Priority , Title , and Description .
Grouping Summary	Indicates the number of topics in each group plus the total number of topics.
Links	Lists the selected topics and any items linked to them.
Summary	Lists each selected topic (or all of them when none are selected). The report gives the Title , Created By , and Created Time fields and the total number of selected topics.

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports and Exporting Data](#)

Related Reference

[Available File Reports](#)

[Available Change Request Reports](#)

[Available Requirement Reports](#)

[Available Task Reports](#)

[Available Folder Reports](#)

[Available Audit Reports](#)

Report Templates

StarTeam provides many reports for each component that you can use as is or customize. Each report is generated using a series of template files that reside in the folder you designated during installation. For example, if you used the default installation path for the Cross-Platform Client on a Windows platform, the Reports folder is `c:\Program Files\Borland\StarTeam Cross-Platform Client 2009\samples\details-templates\`.

To customize the reports, you can open and edit the report templates in any text editor, Microsoft's Developer Studio, or HTML tool.

Template Files

The template files provided for each report are listed below.

Report Name	Template Files
Default	<pre><component>Default.Title <component>Default.Name <component>Default.GrpInfo <component>Default.Group1 <component>Default.EndReport</pre>
Detail	<pre><component>Detail.Title <component>Detail.Name <component>Detail.GrpInfo <component>Detail.Group1 <component>Detail.EndReport</pre>
Detail with Description	<pre><component>Detail with Description.Title <component>Detail with Description.Name <component>Detail with Description.GrpInfo <component>Detail with Description.Group1 <component>Detail with Description.Group2 <component>Detail with Description.EndReport</pre>
Grouping Summary	<pre><component>Grouping Summary.Title <component>Grouping Summary.Name <component>Grouping Summary.GroupSummary <component>Grouping Summary.EndReport</pre>
History	<pre><component>History.Title <component>History.Name <component>History.GrpInfo <component>History.Group1 <component>History.Group2 <component>History.EndReport</pre>
Links	<pre><component>History.Title <component>Links.Name <component>Links.GrpInfo <component>Links.Group1</pre>

```
<component>Links.Group2
<component>Links.EndReport
```

Summary

```
<component>Summary.Title
<component>Summary.Name
<component>Summary.GrpInfo
<component>Summary.Group1
<component>Summary.EndReport
```

Summary with Description

```
<component>Summary with description.Title
<component>Summary with description.Name
<component>Summary with description.GrpInfo
<component>Summary with description.Group1
<component>Summary with description.EndReport
```

Understanding the Template Files

The report template file names are composed as `<component><name>.<type>`. The table below describes each template file type.

Template File Type	Description
<code>.Title</code>	In the <code>.Title</code> templates, the report title is represented with the <code>~~ReportTitle~~</code> tag. In these templates, you might want to add your company's logo, change the font or size, etc.
<code>.GrpInfo</code>	The <code>.GrpInfo</code> templates contain information describing the group, which is represented as the <code>~~GroupingInfo~~</code> tag.
<code>.Group1..9</code>	The <code>.Groupx</code> files are processed in the numerical order in which they are found. The numbers do not have to be contiguous. The application searches for <code>.Group1</code> through <code>.Group9</code> as it creates a record for each item in the report. The names of the fields in the component are individually specified within the double tildes (<code>~~</code>). You must use the correct SQL name. The fields' data will replace the SQL name and the double tildes. To appear in the report, all the specified fields must be displayed in the upper pane at the time the report is generated. Otherwise, those fields are skipped.
<code>.GroupSummary</code>	The <code>.GroupSummary</code> templates are used when totalling the items in a group. The group is represented with the <code>~~GroupingInfo~~</code> tag.
<code>.EndReport</code>	This template ends your report. It may include the following total tags: <code>~~TotalHistoryCount~~</code> <code>~~TotalRecordCount~~</code> <code>~~TotalLinkCount~~</code>

Related Concepts

[Reports](#)

Related Procedures

[Creating Reports and Exporting Data](#)

Related Reference

[Available File Reports](#)

[Available Change Request Reports](#)

[Available Requirement Reports](#)

[Available Task Reports](#)

[Available Topic Reports](#)

[Available Folder Reports](#)

[Available Audit Reports](#)

Personal Options Settings

This section contains the documentation that explains all the settings in the Personal Options dialog box.

In This Section

- [Workspace Options \(Personal Options Dialog Box\)](#)
- [StarTeamMPX Options \(Personal Options Dialog Box\)](#)
- [File Options \(Personal Options Dialog Box\)](#)
- [Change Request Options \(Personal Options Dialog Box\)](#)
- [Requirements Options \(Personal Options Dialog Box\)](#)
- [Task Options \(Personal Options Dialog Box\)](#)
- [Topic Options \(Personal Options Dialog Box\)](#)
- [Folder Options \(Personal Options Dialog Box\)](#)

Workspace Options (Personal Options Dialog Box)

Tools ► Personal Options ► Workspace

The **Workspace** personal options allow you to select a variety of options that affect the way your workspace operates.

Item	Description
Delete	Displays a Confirm dialog box for deletions.
Move/Share	Displays a Confirm dialog box for move and share operations.
Warnings	Displays a Confirm dialog box for warnings.
Toolbars	Displays Toolbars .
Status bar	Displays the Status Bar .
Custom tools	Displays custom tools created as part of StarTeam Extensions. If no custom tools are configured, clear the Custom tools check box to prevent custom tools from attempting to load with each view window.
Show history times as UTC	Displays time stamps in the Time column of the history pane in UTC times. UTC times end in "Z" to differentiate them from local times (Z stands for the "zero meridian", which goes through Greenwich, England.). Displays time stamps in local time when unchecked.
Sort view labels by name	Automatically sorts view labels alphabetically.
Restore folder selection on tab change	Returns StarTeam to the last folder that was selected for a specific component tab. If you have not selected a particular tab in this session, StarTeam automatically selects the root folder for the view.
Reset scope to local on folder change	Resets the All Descendants button on the toolbar every time you change a folder. This saves you the time StarTeam would take to scan items. If unchecked, you must select the button manually.
Maintain group state on folder/scope change	Keeps your place in the upper pane even when you change folders or reset the All Descendants button. If unchecked, all groups collapse when you make a folder or scope change.
Automatic refresh with maximum delay of ____ minutes	Specifies the maximum number of minutes between refreshes of your project. This refresh is the equivalent of pressing SHIFT + F5 and updates the Folder Tree and the upper pane. It happens every X minutes unless an operation by the user such as pressing SHIFT + F5 forces a refresh and resets the timer. Automatic refresh is designed to perform even if the client is minimized. When Automatic refresh enabled, if the network is down, Automatic refresh will attempt to refresh the data in the client resulting in a connection error.
Restore shortcuts at startup	Reopen at startup any views that were left open when you exited the application the previous time.
Report output path	Specifies the name and path to which your reports should be sent
Log output path	Specifies the name and path for your StarTeam.Log files. The StarTeam.Log file contains data about operations sent from your workstation to one or more servers, depending on what project views you have open. The data includes the name of the project so that you can isolate data for a particular server when necessary.
Log errors	Records errors that occur while you are using the client application. The errors log lists the date and time you started your server configuration and any errors or failed operations between the server and client. The application identifies each failed operation by an internal ID and provides an explanation. For example, you might see: <code>...Operation 40956 failed: TCP/IP Socket Error 10054:....</code> If you are logging both errors and operations, the

	<p>application also logs the operation that the server was performing at the time of the error.</p>
Log operations and events that take at least ____ (milliseconds)	<p>Specifies that StarTeam should record file operations and/or events that take at least the specified number of milliseconds, and should send them to the <code>.log</code> files. The milliseconds time setting stops the log from filling up with operations and events of little importance. The default, 10 milliseconds, is a reasonable setting.</p> <p>The Summary option includes a breakdown of the time spent on the client and the server for each operation, and the Details option lists the server commands along with the summary.</p> <p>This option records information on the date, time, and UI Operation number for each command executed by your workstation. Operations can be executed on either the server or the client.</p>
Log StarTeamMPX Server events	<p>Specifies that StarTeamMPX server events should be sent to the <code>.log</code> files. The log identifies the time and date on which a StarTeamMPX event (an automatic refresh or file status update) took place. The log prefaces a StarTeamMPX event as “Statistics for Events” and uses internal IDs and brief explanations to identify the server event.</p> <p>The following example describes a status change for a file: <code>...Statistics for Events /1b21dd1-e208-51ea-01b2-1dd1e20851ea/Object/File/ Modify.</code></p> <p>You can log StarTeamMPX events only if you check Enable StarTeamMPX on the StarTeamMPX tab in the Personal Options dialog box. For StarTeamMPX related operations, any changes you make on the Workspace tab do not apply to projects already open. However, the application will log StarTeamMPX events for any projects you open after checking this option.</p>

Advanced Options

Component Order	<p>Specifies the display order for the component tabs in the application. The tab display to the left of the Component Order button displays the tabs in the order you chose. Important: It is necessary to close and reopen the project for changes to the component tab order to take effect.</p> <p>Note: Never list the Audit component tab first in the component list. Doing so greatly impacts performance.</p>
Look and feel	<p>Changes the look and feel of the application.</p> <p>Note: This option is not available in the StarTeam Visual Studio Integration.</p>
Font adjustment	<p>Increases or decreases the application font size.</p> <p>Note: This option is not available in the StarTeam Visual Studio Integration.</p>
Application Instance	<p>Specifies the default behavior for opening another instance of StarTeam when using a shortcut or URL.</p> <p>Using a URL or shortcut defaults to using an open instance of the StarTeam client to open. However if the client has a dialog open inside some view (or some other modal operation in process), StarTeam cannot use open a project view in that client. In such situations, when you click a URL from email (for example) or a shortcut on your desktop, or a <code>.vcmx</code> file attached to a merge process task, a new instance of StarTeam is opened ONLY if the option selected from the Application Instance group box allows it. If the option is not selected, and you try to open a shortcut, you may get an error saying your StarTeam Client is busy.</p>

Note: This option is not available in the StarTeam Visual Studio Integration.

Prompt when attempting to run another application instance	Sets the default to prompt you whether to open another instance of StarTeam.
Always run another application instance.	Sets the default to always open another instance of StarTeam.
Never run another application instance.	Sets the default to never open another instance of StarTeam.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Customizing Personal Options](#)

[Displaying and Customizing Logging Options](#)

Related Reference

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Folder Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

StarTeamMPX Options (Personal Options Dialog Box)

Tools ► Personal Options ► StarTeamMPX

The options on this page enable and configure StarTeamMPX and Cache Agent on your workstation.

Item	Description
Enable StarTeamMPX	Enables your workstation to use StarTeamMPX if it is available on the server. Changing this check box does not affect open projects. StarTeamMPX is enabled by default.
Automatic refresh with	<p>Enables automatic refresh of the application window by way of MPX, with options for setting the minimum and maximum delay times between refreshes. The default minimum is 30 seconds, and the default maximum is 0–seconds. If this option is unchecked, you must refresh manually (SHIFT+F5.)</p> <p>When Automatic Refresh is enabled, after every change to the view, StarTeam waits a minimum number of seconds before refreshing. That means that if changes are infrequent, the application performs a refresh almost immediately. However, if changes are frequent, the minimum refresh timer is constantly being reset and never reaches the number of seconds set for a refresh. In such cases, the next refresh occurs when the maximum number of seconds between refreshes forces a refresh.</p> <p>Automatic refresh is designed to perform even if the client is minimized. When Automatic refresh enabled, if the network is down, Automatic refresh will attempt to refresh the data in the client resulting in a connection error.</p>
Enable StarTeamMPX Cache Agent	Uses a StarTeamMPX Cache Agent when checking out files.
Use Cache Agent at	Designates a specific Cache Agent to use by IP address and port number. If you specify a Cache Agent in this option, you do not have to have the StarTeam Server MPX-enabled at the time you run the Bulk Check-out (BCO) utility.
Automatically locate the closest Cache Agent for StarTeamMPX Server	Locates the network nearest Cache Agent automatically, but only if the server is MPX-enabled.
Maximum request threads	Specifies the maximum number of request threads allowed. The default is 2, and 2 to 3 should be adequate for most of your needs.
Use Cache Agent for	Specifies whether the Cache Agent will be used for file caching (File content) and/or object caching (Item properties).

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Customizing Personal Options](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Folder Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

File Options (Personal Options Dialog Box)

Tools ► Personal Options ► File

Use this page of the **Personal Options** dialog box to set file options.

Item	Description
Use last modification time for check-out files	Uses the same time for each checked-out file as the time stamp of the revision being checked out. Otherwise, the time stamp used for the checked-out file is the current time (the time check-out occurs.)
Always pop-up merge utility	Opens the merge utility to display the merged file even when there are no conflicts.
Pop-up merge utility in case of conflicts only	Opens the merge utility only if the merged file contains conflicts. If unchecked, files will be checked in and out automatically.
Exclusively lock files on check-out	Sets the default Lock Status option to Exclusive in the Check Out dialog box . Otherwise, the default is Keep Current .
Clear file locks on check-in	Sets the default Lock Status option to Unlock in the Check Out dialog box. Otherwise, the default is Keep Current .
Use non-exclusive locks in integration	Creates a non-exclusive lock when locking a file from the application integration— that is, a lock that allows others to check in the file. Using non-exclusive locks also allows more than one person to edit a file at one time. If team members are not editing the same lines of the file, the merged file usually has no conflicts.
Mark unlocked working files read-only	Sets working copies of files that you have not locked to read-only when you add files, check in files, check out files, or unlock files. If this check box is selected, only locked files can be edited.
Automatic EOL conversion for check-out operations	<p>Performs an automatic EOL conversion on check-out operations. If checked, also select the operating system on which you are working: Windows (CR-LF), Unix(LF), or Mac(R).</p> <p>Files can be checked out in LF format on every platform, regardless of specific options.</p> <p>The EOL Format property can be set in the Cross-Platform Client in the Add/Check-in and File Properties dialog boxes.</p> <p>The default for automatic EOL conversion for check-out operations is “checked” if the user does not have that option defined already. Users that upgrade to 2009 should check that option to be sure they have it set correctly given the new EOL Format changes.</p> <p>The EOL Property values are:</p> <p><i>Undefined</i> (null in the SDK): Used for files added before StarTeam 2009.</p> <p><i>Client Defined</i>: Causes workstation default or per-checkout EOL conversion option to be used.</p> <p><i>Fixed CR</i>, <i>Fixed LF</i>, and <i>Fixed CRLF</i>: Causes this EOL format to be used always; the workstation/check-out conversion option is ignored.</p> <p>Note: Once EOL Format is defined, Update Status works for all text files, regardless of what EOL format was used when they were checked-out. For compatibility with older Clients, if check-out "EOL conversion" is not requested, and EOL Format is Undefined, files are still checked out with the EOL convention with which they were added to the Server.</p>

Use file checksums (MD5) to calculate status	Uses the checksum instead of the file time stamp and size to compute the Status field when the application is refreshed. Using the checksum provides a more accurate status value than the time stamp, but takes longer. If unchecked, the application uses the time stamp and size.
Default file encoding	Specifies the default code page to be used for keyword expansion by choosing a default file encoding from the drop-down list.
File status repository default	<p>Indicates where you want file status information stored, either in a central repository location on your workstation or in a child folder (named <code>.sbas</code>) of each working folder.</p> <p>If you select Central, you can enter or browse for a location on your computer other than the default central repository location. Whenever you make a change to a file in the working folder, the status for that file is undated only on your computer in the specified location. Everyone else sees the status Unknown for that file. Over time, all the files may have been changed, and the statuses can become Unknown for all users of all files.</p> <p>The Per-folder option is most useful in the special case where multiple users are sharing a working folder, for example, on a shared network drive. For example, suppose several users all check files in and out of a shared working folder. If these users have set the central repository option for file statuses, the statuses are stored on each of their computers. Whenever a user makes a change to a file in the working folder, the status for that file is undated only on that user's computer. Everyone else sees the status Unknown for that file. Over time, all the files may have been changed, and the statuses can become Unknown for all users of all files. Using the per-folder option causes the statuses to be updated within the working folder itself. Everyone has access to those status changes and Unknown statuses do not occur.</p> <p>See "Controlling How File Status Information is Stored" in the links below.</p>
Purge	Opens the Status Repository Cleanup dialog box where you can remove file status data from the workstation status repository.
Default	Resets the Central repository location to the default setting
Display template	<p>Specifies a special template used to generate an HTML representation of an item when the item's URL is copied to the Clipboard. With no format, there is a default HTML representation that specifies the type of item and identifies it by the filename, for example, <code>buildinfo.properties</code>. When the text is generated from the template, the specified property values are substituted for the variables in <code>~~*~~</code>. The variables may be referenced by the same names used in report templates, as well as by the display name of the property. When using the display name, you can omit spaces, and case will be ignored. For example, if you use the following sample template for a file: <code>~~FolderPath~~:~~Name~~</code>, the HTML representation will be the HTML representation will be the StarTeam path to the selected file: <code>StarTeam\buildinfo.properties</code>. This template is a superset of that used by the Report feature of the client.</p>
Generate ID-based URLs	<p>Specifies the URL by ID rather than by name. For example, an ID-based URL would be <code>starteam://hostname:49201/12;ns=Project;scheme=id</code>, while a name-based URL would be <code>starteam://hostname:49201/myproject</code>.</p>

Alternate Applications	<p>Note: Folders always use an ID-based URL.</p> <p>Opens the Alternate Applications dialog box where you can specify an alternate editor, merge utility, and comparison utility to use in the application if you don't want to use the default tools for those functions. Includes fields for specifying options to use with the applications.</p>
Open With...	<p>Enables you to provide a command on a non-Windows system that will display at least one type of files and folders. The command should consist of the path to an application and the command-line options for which the application for which the application can substitute the selected file. The application runs this command whenever you do one of the following: Double-click a file or folder in the item list, double-click an attachment, or generate and open a report.</p> <p>The following command is suggested: <code>netscape -remote "openFile(\$file)"</code> because Netscape can handle many different media types, such as image files, text files, and HTML.</p>
Merge Utility Options	<p>Use the following command-line options to represent files sent by StarTeam Cross-Platform Client to the alternate merge utility.</p> <ul style="list-style-type: none"> - <code>\$branchtip</code> A place holder for the path to the tip revision of the file to be merged. - <code>\$usertip</code> A place holder for the path to the local working file to be merged. - <code>\$basefile</code> A place holder for the path to the common ancestor for the <code>\$branchtip</code> and <code>\$usertip</code> files. - <code>\$resultfile</code> A place holder for the path to the file that will store the output from the merged file.
Compare Utility Options	<p>Use the following command-line options to represent files sent by StarTeam Cross-Platform Client to the alternate compare utility.</p> <ul style="list-style-type: none"> - <code>\$file1</code> A place holder for the path to the first of the two files to be compared. - <code>\$file2</code> A place holder for the path to the second of the two files to be compared.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Customizing Personal Options](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[Folder Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

[Report Templates](#)

Change Request Options (Personal Options Dialog Box)

Tools ► Personal Options ► Change Request

The change request options enable you to specify the criteria that the application uses to determine whether a change request has been read. You can also indicate how often the application should search for new change requests and how change request locking issues should be handled.

Item	Description
When change request is selected	Marks a change request read as soon as you select it. An unread change request is displayed with bold font. A read change request is displayed with regular font.
When selected for ____ seconds	Marks a change request read after it has been selected for the specified number of seconds. The range is from 15 to 9999 seconds.
Only when manually marked as read	Marks a selected change request read when you choose Change Request ► Mark as Read . Note: Change requests are always marked as read when you display their properties.
Check for new or modified change requests	Checks for new or modified change requests at regular intervals and lets you know that you have become responsible for new change requests. If this option is checked, you must also specify the number of minutes between checks in the Interval option below. When unchecked, the application does not place an icon in the system tray for a new change request.
Interval (in minutes)	Specifies the number of minutes between automatic checks for new or modified change requests. The default is 10 minutes.
Exclusively lock change request during edit	Locks a change request when you open its Properties dialog box for editing. If unchecked, the application does not lock change requests when you open its Properties dialog box.
Clear manually locked change requests after edit	Unlocks a locked change request after you have edited its properties, and clicked OK to create a new revision. If unchecked, the application does not remove the locks.
Display template	Specifies a special template used to generate an HTML representation of an item when the item's URL is copied to the Clipboard. With no format, there is a default HTML representation that specifies the type of item and identifies it by name and number, for example, Change Request #39,849 . When the text is generated from the template, the specified property values are substituted for the variables in <code>~~*~~</code> . The variables may be referenced by the same names used in report templates, as well as by the display name of the property. When using the display name, you can omit spaces, and case will be ignored. For example, if you use the following sample template for a file: <code>Change Request:~~CR Number~~:~~CreatedBy~~</code> , the HTML representation will be the <code>Change Request:38,849:Tom Smith</code> . This template is a superset of that used by the Report feature of the client.
Generate ID-based URLs	Specifies the URL by ID rather than by name. For example, an ID-based URL would be <code>starteam://hostname:49201/12;ns=Project;scheme=id</code> , while a namebased URL would be <code>starteam://hostname:49201/myproject</code> . Note: Folders always use an ID-based URL.

Note: If you do not select either of the locking options, opening a change request will not lock it; you must manually lock and unlock it. If you select the **Exclusively Lock** option only, change requests that are not already locked become locked when you open them and unlocked when you click **Cancel** or **OK**. If you select the **Clear change request Locks** option only, any change request that you have locked manually becomes unlocked when you click **OK** to create a new revision. If you select both options, you can lock change requests manually or by opening them. These change requests become unlocked when you click **OK** to create new revisions or (if they were not locked prior to being opened) when you click **Cancel**.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Customizing Personal Options](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Folder Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

[Report Templates](#)

Requirements Options (Personal Options Dialog Box)

Tools ► Personal Options ► Requirement

The requirement options enable you to specify the criteria that the application uses to determine whether a requirement has been read. You can also indicate how often the application should search for new requirements and how requirement locking issues should be handled.

Item	Description
When requirement is selected	Marks a requirement read as soon as you select it. An unread requirement is displayed with bold font. A read requirement is displayed with regular font.
When selected for ____ seconds	Marks a requirement read after it has been selected for the specified number of seconds. The range is from 15 to 9999 seconds.
Only when manually marked as read	Marks a selected requirement read when you choose requirement ►Mark as Read . Note: Requirements are always marked as read when you display their properties.
Check for new or modified requirements	Checks for new or modified requirements at regular intervals and lets you know that you have become responsible for new requirements. If this option is checked, you must also specify the number of minutes between checks in the Interval option below. When unchecked, the application does not place an icon in the system tray for a new requirement.
Interval (in minutes)	Specifies the number of minutes between automatic checks for new or modified requirements. The default is 10 minutes.
Exclusively lock requirement during edit	Locks a requirement when you open its Properties dialog box for editing. If unchecked, the application does not lock requirements when you open its Properties dialog box.
Clear manually locked requirements after edit	Unlocks a locked requirement after you have edited its properties, and clicked OK to create a new revision. If unchecked, the application does not remove the locks.
Display template	Specifies a special template used to generate an HTML representation of an item when the item's URL is copied to the Clipboard. With no format, there is a default HTML representation that specifies the type of item and identifies it by name and number. For example, <code>Requirement #34,132</code> . When the text is generated from the template, the specified property values are substituted for the variables in <code>~~*~~</code> . The variables may be referenced by the same names used in report templates, as well as by the display name of the property. When using the display name, you can omit spaces, and case will be ignored. For example, if you use the following sample template for a file: <code>Requirement #~~Number~~: ~~Status~~</code> , the HTML representation will be <code>Requirement #34,132: Submitted</code> . This template is a superset of that used by the Report feature of the client.
Generate ID-based URLs	Specifies the URL by ID rather than by name. For example, an ID-based URL would be <code>starteam://hostname:49201/12;ns=Project;scheme=id</code> , while a namebased URL would be <code>starteam://hostname:49201/myproject</code> . Note: Folders always use an ID-based URL.

Note: If you do not select either of the locking options, opening a requirement will not lock it; you must manually lock and unlock it. If you select the **Exclusively Lock** option only, requirements that are not already locked become locked when you open them and unlocked when you click **Cancel** or **OK**. If you select the **Clear Requirement Locks** option only, any requirement that you have locked manually becomes unlocked when you click **OK** to create a new revision. If you select both options, you can lock requirements manually or by opening them. These requirements become unlocked when you click **OK** to create new revisions or (if they were not locked prior to being opened) when you click **Cancel**.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Customizing Personal Options](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Folder Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

[Report Templates](#)

Task Options (Personal Options Dialog Box)

Tools ► Personal Options ► Task

The task options enable you to specify the criteria that the application uses to determine whether a task has been read. You can also indicate how often the application should search for new tasks and how task locking issues should be handled.

Item	Description
When task is selected	Marks a task read as soon as you select it. An unread task is displayed with bold font. A read task is displayed with regular font.
When selected for ____ seconds	Marks a task read after it has been selected for the specified number of seconds. The range is from 15 to 9999 seconds.
Only when manually marked as read	Marks a selected task read when you choose task ► Mark as Read . Note: Tasks are always marked as read when you display their properties.
Check for new or modified tasks	Checks for new or modified tasks at regular intervals and lets you know that you have become responsible for new tasks. If this option is checked, you must also specify the number of minutes between checks in the Interval option below. When unchecked, the application does not place an icon in the system tray for a new task.
Interval (in minutes)	Specifies the number of minutes between automatic checks for new or modified tasks. The default is 10 minutes.
Exclusively lock task during edit	Locks a task when you open its Properties dialog box for editing. If unchecked, the application does not lock tasks when you open its Properties dialog box.
Clear manually locked tasks after edit	Unlocks a locked task after you have edited its properties, and clicked OK to create a new revision. If unchecked, the application does not remove the locks.
Display template	Specifies a special template used to generate an HTML representation of an item when the item's URL is copied to the Clipboard. With no format, there is a default HTML representation that specifies the type of item and identifies it by name and number. For example, <code>Task #1,456</code> . When the text is generated from the template, the specified property values are substituted for the variables in <code>~~*~~</code> . The variables may be referenced by the same names used in report templates, as well as by the display name of the property. When using the display name, you can omit spaces, and case will be ignored. For example, if you use the following sample template for a file: <code>Task #~~Task Number~~:~~Status~~:~~Responsibility~~</code> , the HTML representation will be <code>Task #1,456:Ready To Start:Tom Smith</code> . This template is a superset of that used by the Report feature of the client.
Generate ID-based URLs	Specifies the URL by ID rather than by name. For example, an ID-based URL would be <code>starteam://hostname:49201/12;ns=Project;scheme=id</code> , while a namebased URL would be <code>starteam://hostname:49201/myproject</code> . Note: Folders always use an ID-based URL.

Note: If you do not select either of the locking options, opening a task will not lock it; you must manually lock and unlock it. If you select the **Exclusively Lock** option only, tasks that are not already locked become locked when you open them and unlocked when you click **Cancel** or **OK**. If you select the **Clear task Locks** option only, any task that you have locked manually becomes unlocked when you click **OK** to create a new revision.

If you select both options, you can lock tasks manually or by opening them. These tasks become unlocked when you click **OK** to create new revisions or (if they were not locked prior to being opened) when you click **Cancel**.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Customizing Personal Options](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Folder Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

[Report Templates](#)

Topic Options (Personal Options Dialog Box)

Tools ▶ Personal Options ▶ Topic

The topic options enable you to specify the criteria that the application uses to determine whether a topic has been read. You can also indicate how often the application should search for new topics and how topic locking issues should be handled.

Item	Description
When topic is selected	Marks a topic read as soon as you select it. An unread topic is displayed with bold font. A read topic is displayed with regular font.
When selected for ____ seconds	Marks a topic read after it has been selected for the specified number of seconds. The range is from 15 to 9999 seconds.
Only when manually marked as read	Marks a selected topic read when you choose topic ▶ Mark as Read . Note: Topics are always marked as read when you display their properties.
Check for new or modified topics	Checks for new or modified topics at regular intervals and lets you know that you have become responsible for new topics. If this option is checked, you must also specify the number of minutes between checks in the Interval option below. When unchecked, the application does not place an icon in the system tray for a new topic.
Interval (in minutes)	Specifies the number of minutes between automatic checks for new or modified topics. The default is 10 minutes.
Exclusively lock topic during edit	Locks a topic when you open its Properties dialog box for editing. If unchecked, the application does not lock topics when you open its Properties dialog box.
Clear manually locked topics after edit	Unlocks a locked topic after you have edited its properties, and clicked OK to create a new revision. If unchecked, the application does not remove the locks.
Display template	Specifies a special template used to generate an HTML representation of an item when the item's URL is copied to the Clipboard. With no format, there is a default HTML representation that specifies the type of item and identifies it by name and number. For example, <code>Topic #34,132</code> . When the text is generated from the template, the specified property values are substituted for the variables in <code>~~*~~</code> . The variables may be referenced by the same names used in report templates, as well as by the display name of the property. When using the display name, you can omit spaces, and case will be ignored. For example, if you use the following sample template for a topic: <code>Topic # ~~topicnumber~~: ~~Title~~, Status - ~~status~~</code> , the HTML representation will be <code>Topic #34,132: Topic Title, Status - Active</code> . This template is a superset of that used by the Report feature of the client.
Generate ID-based URLs	Specifies the URL by ID rather than by name. For example, an ID-based URL would be <code>starteam://hostname:49201/12;ns=Project;scheme=id</code> , while a namebased URL would be <code>starteam://hostname:49201/myproject</code> . Note: Folders always use an ID-based URL.

Note: If you do not select either of the locking options, opening a topic will not lock it; you must manually lock and unlock it. If you select the **Exclusively Lock** option only, topics that are not already locked become locked when you open them and unlocked when you click **Cancel** or **OK**. If you select the **Clear topic Locks** option

only, any topic that you have locked manually becomes unlocked when you click **OK** to create a new revision. If you select both options, you can lock topics manually or by opening them. These topics become unlocked when you click **OK** to create new revisions or (if they were not locked prior to being opened) when you click **Cancel**.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Customizing Personal Options](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Folder Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Report Templates](#)

Folder Options (Personal Options Dialog Box)

Tools ► Personal Options ► Folder

The folder options enable you to specify the criteria that the application uses to determine whether a folder has been read. You can also indicate how often the application should search for new folders and how folder locking issues should be handled.

Item	Description
Exclusively Lock Folder During Edit	Locks a folder when you open its Properties dialog box for editing. If unchecked, the application does not lock folders when you open its Properties dialog box.
Clear Manually Locked Folders After Edit	Unlocks a locked folder after you have edited its properties, and clicked OK to create a new revision. If unchecked, the application does not remove the locks.
Display template	Specifies a special template used to generate an HTML representation of an item when the item's URL is copied to the Clipboard. With no format, there is a default HTML representation that specifies the type of item and identifies it by the folder name. For example, ReadMe . When the text is generated from the template, the specified property values are substituted for the variables in <code>~~*~~</code> . The variables may be referenced by the same names used in report templates, as well as by the display name of the property. When using the display name, you can omit spaces, and case will be ignored. For example, if you use the following sample template for a file: <code>~~FolderPath~~:~~Name~~</code> , the HTML representation will be the StarTeam path to the selected folder: <code>SampleProject\ReadMe</code> . This template is a super-set of that used by the Report feature of the client.
Generate ID-based URLs	Specifies the URL by ID rather than by name. For example, an ID-based URL would be <code>starteam://hostname:49201/12;ns=Project;scheme=id</code> , while a name-based URL would be <code>starteam://hostname:49201/myproject</code> . Note: Folders always use an ID-based URL.
Show Not-in-View Folders By Default	Checks the Show Not-in-View Folders on the Folder Tree menu to set it to be on by default. (Changing this check box does not affect projects that are already open.)
Open With...	Enables you to provide a command on a non-Windows system that will display at least one type of files and folders. The command should consist of the path to an application and the command-line options for which the application for which the application can substitute the selected file. The application runs this command whenever you do one of the following: Double-click a file or folder in the item list, double-click an attachment, or generate and open a report. The following command is suggested: <code>netscape -remote "openFile(\$file)"</code> because Netscape can handle many different media types, such as image files, text files, and HTML.

Note: If you do not select either of the locking options, opening a folder will not lock it; you must manually lock and unlock it. If you select the **Exclusively Lock** option only, folders that are not already locked become locked when you open them and unlocked when you click **Cancel** or **OK**. If you select the **Clear folder Locks** option only, any folder that you have locked manually becomes unlocked when you click **OK** to create a new revision. If you select both options, you can lock folders manually or by opening them. These folders become unlocked

when you click **OK** to create new revisions or (if they were not locked prior to being opened) when you click **Cancel**.

Related Concepts

[Personal Options](#)

Related Procedures

[Controlling How File Status Information is Stored](#)

[Customizing Personal Options](#)

Related Reference

[Workspace Options \(Personal Options Dialog Box\)](#)

[StarTeamMPX Options \(Personal Options Dialog Box\)](#)

[File Options \(Personal Options Dialog Box\)](#)

[Change Request Options \(Personal Options Dialog Box\)](#)

[Requirements Options \(Personal Options Dialog Box\)](#)

[Task Options \(Personal Options Dialog Box\)](#)

[Topic Options \(Personal Options Dialog Box\)](#)

[Report Templates](#)

Views

This section provides reference information on creating and configuring views.

In This Section

[View Type Options and Settings](#)

This table describes the option settings in the New View Wizard for creating different types of views.

[View Configuration Options](#)

This table describes the view configuration options available in the New View Wizard.

View Type Options and Settings

The table below lists the settings that must be selected in the New View Wizard to create the different views.

Desired Characteristics for New View	Options to Set
<p><i>Branching: Branch All (Not Floating)</i></p> <p>New Items:</p> <p>New items in the child view do not appear in the parent view; new items in the parent view do not appear in the child view.</p> <p>Existing Items:</p> <p>Existing items in the child view are the same as in parent view at the time of configuration, until the item in the child view branches. Changes cannot be made to an item in the child view if the change does not result in branching.</p> <p>Item Behavior:</p> <p>Branch on Change check box is enabled and selected for all items that can branch.</p> <p>Any change to a child item that can branch results in the branching of that item, unless its Branch on Change check box has been cleared. After branching, the check box is disabled.</p>	<p>View Type:</p> <p>Branch All</p> <p>Root Folder:</p> <p>Selected from parent view</p> <p>Working Folder:</p> <p>Should be different from that of the parent, to avoid conflicts</p> <p>Configuration:</p> <p>Other than floating (Labeled, Promotion state, or As of specific date)</p> <p>Note: Addressed In Build field when the child view is created, "Next Build" will not be replaced by a build label until the change request branches.</p>
<p><i>Read-only Reference View (Frozen)</i></p> <p>New Items:</p> <p>New items cannot be added to child view; new items in parent view appear in child view.</p> <p>Existing Items:</p> <p>Existing items are the same in the child view and the parent view; they can be changed only from the parent view.</p> <p>Item Behavior:</p> <p>Branch on Change has the same setting as the parent item, but is irrelevant; no change can be made..</p>	<p>View Type:</p> <p>Read-only Reference</p> <p>Root Folder:</p> <p>Selected from parent view</p> <p>Working Folder:</p> <p>Usually the same as that of the parent view</p> <p>Configuration:</p> <p>Other than Floating (Labeled, Promotion State, or AS of Specified Date)</p> <p>These views can be rolled back.</p>
<p><i>Non-derived View (also called Blank Branching View)</i></p> <p>New Items:</p> <p>New items in child view do not appear in the parent view; new items in the parent view do not appear in the child view.</p> <p>Existing Items:</p> <p>Existing items in the parent view do not appear in the child view.</p> <p>Item Behavior:</p> <p>Branch on Change check box disabled.</p>	<p>View Type:</p> <p>Non-Derived</p> <p>Root Folder:</p> <p>N/A</p> <p>Working Folder:</p> <p>Should be different from that of the parent to avoid conflicts.</p> <p>Configuration:</p> <p>N/A</p>
<p><i>Reference View (Also called Read/Write Reference View)</i></p> <p>New Items:</p> <p>New items in the child view appear in both views; new items in the parent view appear in both views if they are in the subset accessed by the child view.</p> <p>Existing Items:</p>	<p>View Type:</p> <p>Reference</p> <p>Root Folder:</p> <p>Selected from parent view</p> <p>Working Folder:</p>

Existing items are the same in the child view and the parent view; they can be changed from either view.

Item Behavior:

Branch on Change has the same setting as the parent item.

Note: Labels created and objects deleted in the child view appear and disappear in the parent view; this is not true for other types of child views.

Branching: Branch None (Not Floating)

Not recommended.

New Items:

New items in the child view do not appear in the parent view; new items in the parent view do not appear in the child view.

Existing Items:

Existing items in the child view are the same as in the parent view at the time of configuration, until the item in the child view branches. Changes cannot be made to an item in the child view if that change does not result in branching.

Item Behavior:

Branch on Change check box enabled, but initially cleared.

No change to a child item that can branch results in branching until the Branch on Change check box is selected. After branching, the box is disabled.

Branching: Branch None (Floating)

Not recommended. Use only when a different set of view labels is needed for the same data.

New Items:

New items in the child view appear in the parent view; new items in the parent view appear in the child view if they are in the subset accessed by the child view. In the child view, new items from the parent have the Branch on Change check box cleared.

Existing Items:

Existing items are the same in the child view as in the parent view; they can be changed in either the parent or child view until the item in the child view branches. However, items deleted from one view are not deleted from the other.

Item Behavior:

Branch on Change check box enabled, but initially cleared

No change to a child item that can branch results in branching until the Branch on Change check box is selected. After branching, the check box is disabled.

Branching: Branch All (Floating)

Not recommended.

New Items:

New items in the child view do not appear in the parent view; new items in the parent view appear in both views if they are in the subset accessed by the child view. In the child view, new

Usually the same as that of the parent view

Configuration:

N/A; always floats.

View Type:

Branch None (an advanced type)

Root Folder:

Selected from parent view

Working Folder:

Should be different from that of the parent, to avoid conflicts Configuration:

Configuration:

Other than floating (Labeled, Promotion State, or As of specific date)

Note: Addressed In Build field when the child view is created, "Next Build" will not be replaced by a build label until the change request branches.

View Type:

Branch None (an advanced type)

Root Folder:

Selected from parent view

Working Folder:

Should be different from that of the parent to avoid conflicts.

Configuration:

Floating

Note: If users perform many move and share operations, using branching, floating views can result in multiple unwanted references to the same folders or items, causing confusion. Also, if a change request has "Next Build" in the Addressed In Build field when the child view is created, "Next Build" will be replaced by the parent's next build label, unless the change request branches in the child view.

View Type:

Branch All, Float (an advanced type)

Root Folder:

Selected from parent view

Working Folder:

items from the parent have the Branch on Change check box selected.

Existing Items:

Changes to existing items in the parent view appear in the child view until the corresponding item in the child view branches. Changes to existing items in the child view can appear in the parent view, but only if the Branch on Change check box for that item is cleared. However, items deleted from one view are not deleted from the other.

Item Behavior:

Branch on Change check box is enabled and initially selected for all items that can branch.

Any change to a child item that can branch results in the branching of that item, unless its Branch on Change check box has been cleared. After branching, the box is disabled.

Branching: Branch All (Floating)

New Items:

New items in the child view do not appear in the parent view; new items in the parent view appear in both views if they are in the subset accessed by the child view. In the child view, new items from the parent have the Branch on Change check box selected.

Existing Items:

Changes to existing items in the parent view appear in the child view until the corresponding item in the child view branches. Changes to existing items in the child view can appear in the parent view, but only if the Branch on Change check box for that item is cleared. However, items deleted from one view are not deleted from the other.

Item Behavior:

Branch on Change check box is enabled and selected for all items that can branch.

Any change to a child item that can branch results in the branching of that item, unless its Branch on Change check box has been cleared. After branching, the box is disabled.

Should be different from that of the parent, to avoid conflicts

Configuration:

Floating

Note: If users are likely to perform many move and share operations, using branching, floating views can result in multiple unwanted references to the same folders or items, causing confusion. Also, if a change request has "Next Build" in the Addressed In Build field when the child view is created, "Next Build" will be replaced by the parent's next build label, unless the change request is first branched.

View Type:

Branch All

Root Folder:

Selected from parent view

Working Folder:

Should be different from that of the parent, to avoid conflicts

Configuration:

Floating

Note: If users are likely to perform many move and share operations, using branching, floating views can result in multiple unwanted references to the same folders or items, causing confusion. Also, if a change request has "Next Build" in the Addressed In Build field when the child view is created, "Next Build" will be replaced by the parent's next build label, unless the change request is first branched.

Related Concepts

[Understanding View Types](#)

[Overview of Views](#)

Related Procedures

[Managing Views](#)

[Creating and Configuring Views](#)

Related Reference

[View Configuration Options](#)

View Configuration Options

This table describes the view configuration options available in the New View Wizard.

Configuration	Description
Floating Configuration	Not recommended. All the items in the new view will be identical to the corresponding items in the current parent view. Changes to an item in the parent view will be made to the corresponding item in the new view until that item branches, while changes to an item in the new view will be reflected in the parent until the item in the new view branches. (In many cases, the first change to that item will result in branching). New items in the parent view will appear in a branching view. However, new items added to a branching view will appear in the parent view only if the new view is the Branch none type.
Labeled Configuration	All the items in the new view will have had the specified label in the parent view. In all cases, the revision of the item to which the label was attached is the tip revision in the new view. This option is disabled if the parent view has no view labels. Changes to the parent view do not affect the new view, including changes to the label upon which the view is based. Unless an item is set to Branch on change in the new view, it will be read-only and you cannot change it.
Promotion State Configuration	All the items in the new view will have been part of the specified promotion state in the parent view. In all cases, the revision of the item that was part of the promotion state is the tip revision in the new view. This option is disabled if the parent view has no promotion states defined for it. Changes to the parent view will not affect the new view, including changes to the promotion state or its assigned label. Unless a specific item is set to Branch on change in the new reference view, it will be read-only and you cannot change it.
Configuration As Of	The new view will contain only the items that existed at the date and time you specify. In all cases, the tip revision of each item in the new view is the revision closest to, but before, the specified time. Changes to the parent view will not affect the new view. Unless a specific item is set to Branch on change in the new view, it will be read-only and you cannot change it.

Related Concepts

[Understanding View Types](#)
[Overview of Views](#)

Related Procedures

[Managing Views](#)
[Creating and Configuring Views](#)

Related Reference

[View Type Options and Settings](#)

Audit Log Events

Events are actions performed on an owner. For example, a file can be checked in or removed from version control. Such events are recorded in the audit log. Most items can be:

- ◆ Added
- ◆ Branched
- ◆ Comment Edited
- ◆ Created
- ◆ Deleted
- ◆ Locked
- ◆ Lock Broken
- ◆ Modified
- ◆ Moved From
- ◆ Moved to
- ◆ Shared
- ◆ Unlocked
- ◆ Converted
- ◆ Edited
- ◆ Item Overwritten (as foreign archive files become native files)
- ◆ Vault
- ◆ Created
- ◆ Modified
- ◆ Deleted
- ◆ Frozen
- ◆ Unfrozen
- ◆ Attached
- ◆ Moved
- ◆ Detached
- ◆ Modified

Table of Common Operations

The following table lists the generic operations that can be performed with each component.

Operation	File	Change Request	Requirement	Task	Topic	Folder	Audit
Moving	Yes	Yes	No	Yes, except when using Microsoft Project Integration	Yes	Yes	No
Drag an item to a new location	No	No	No	No	No	Moving a folder moves its contents, child folders, and their contents.	No
Creating shortcuts to items	Yes	Yes	Yes	Yes	Yes	Yes	No
Copying items to a third-party application via a URL	Yes	Yes	Yes	Yes	Yes	Yes	No
Sharing <code>Ctrl+drag</code> item to a new location	Yes	Yes	Yes	Yes	Yes	Yes	No
Branching behavior	Yes	Yes	No	No	No	Yes	No
Configuring to or freezing at a point in the past	Yes	Yes	Yes	Yes	Yes	Yes	No
Locking	Yes	Yes	Yes	Yes	Yes	No	No
Comparing properties of two items of the same type	Yes	Yes	Yes	Yes	Yes	No	No
Comparing properties of two revisions	Yes	Yes	Yes	Yes	Yes	Yes	No
Review revision history	Yes	Yes	Yes	Yes	Yes	Yes	No
Viewing revision properties	Yes	Yes	Yes	Yes	Yes	Yes	No
Editing revision comments	Yes	Yes	Yes	Yes	Yes	Yes	No
Merging revisions	Yes, using Visual Merge	No, except as a part of merging views	No, except as a part of merging views	No, except as a part of merging views	No, except as a part of merging views	No, except as a part of merging views, which is often done by an administrator	No
Finding based on field content	Yes	Yes	Yes	Yes	Yes	No	Yes
Selecting by query	Yes	Yes	Yes	Yes	Yes	No	Yes

Selecting by label	Yes	Yes	Yes	Yes	Yes	No	No
Label revisions	Yes	Yes	Yes	Yes	Yes	Yes	No
Viewing references	Yes	Yes	Yes	Yes	Yes	Yes	No
Linking to folders and items	Yes	Yes	Yes	Yes	Yes	Yes	No
Printing a default report for selected items	Yes	Yes	Yes	Yes	Yes	No	No
Sending items as email	No	Yes	Yes	Yes	Yes	No	Yes
Receiving email notification about changes (when notification is enabled by administrator)	No	Yes; changes in responsibility only	Yes; all changes in items for which you are responsible	Yes; all changes in items for which you are responsible	Yes; changes in items for which you are a recipient	No	No
Controlling system tray notification	No	Yes	Yes	Yes	Yes	No	No
Marking items as read/unread	No	Yes; you can also mark trees as read/unread	Yes; you can also mark trees as read/unread	Yes; you can also mark trees as read/unread	Yes; you can also mark trees as read/unread	No	No
Flagging items	Yes	Yes	Yes	Yes	Yes	No	No
Deleting	Yes	Yes	Yes	Yes	Yes	Yes	No
Setting access rights. Normally, this function is performed by Administrators.	Yes	Yes	Yes	Yes	Yes	Yes	No
Creating reports	Yes	Yes	Yes	Yes	Yes	No	Yes
Creating charts	Yes	Yes	Yes	Yes	Yes	No	Yes

Table of StarTeam Keywords

By enabling keyword expansion for a project, you can embed keywords within text files. These keywords are automatically expanded during file check-outs, to provide file and revision information within the file. You should use only one keyword per line.

Item	Description
<code>\$Author\$</code>	User who checked in the revision.
<code>\$Date\$</code> and <code>\$DateUTC\$</code>	Date and time stamps for the revision. <code>\$DateUTC\$</code> is the same as <code>\$Date\$</code> except that a UTC time replaces the local time. UTC times end in a "Z," which makes them readily identifiable.
<code>\$Header\$</code> and <code>\$HeaderUTC\$</code>	<p>Combinations of Workfile, Revision, Date, and Author. <code>\$HeaderUTC\$</code> is the same as <code>\$Header\$</code> except that a UTC time replaces the local time. UTC times end in a "Z" which makes them readily identifiable.</p> <p>For Java users, <code>\$Header\$</code> can cause problems if <code>\u</code> (for unicode) appears in the expanded header. For example, suppose that a file named <code>foo.java</code> is stored in <code>D:\util</code>. The first time you compile it with <code>\$Header\$</code>, the header is expanded to:</p> <pre>\$Header: D:\util\foo.java, 1, 7/27/99 11:05:48 AM, StarTeam Server Administrator\$</pre> <p>Even though this header is contained in a Java comment, the Java compiler always looks for <code>\u</code>. The second time you compile <code>foo.java</code>, a compiler error occurs.</p>
<code>\$History\$</code>	<p>Added to the source file, typically within a comment field. The <code>\$History\$</code> keyword creates a history record for the latest (tip) revision and places the information after the keyword. For example, after the file is checked-out for the first time, the file would contain the following:</p> <pre>// ... // \$History // 1 YourProject 1.0 2005-11-19 00:06:57Z Joe Smith // This is a revision comment. // \$</pre> <p>StarTeam adds the history information to the file and places it outside of the "\$" signs so that it becomes a versioned part of the file. Since the history records become part of the versioned file content, you can delete extra or excessive history records at any time. If this file is modified and checked-in, any subsequent check-out adds one additional record immediately after the <code>\$History\$</code> keyword:</p> <pre>// \$History // 2 YourProject 1.0 2006-06-01 00:06:57Z Joe Smith</pre>

```
// This is a another revision comment.
// $
// 1 YourProject 1.0 2005-11-19 00:06:57Z
Joe Smith
// This is a revision comment.
```

StarTeam uses the MD5 and local file timestamp of the post-expanded file for sync records. Accordingly, when you check-out a tip revision, StarTeam reports a *current* status for the file even though a compare tool would show an additional history record in the locally-stored file as compared to the version of the file saved on the Server.

Important: Do not delete the standalone `// $` line. StarTeam places the most recent historical revision information within the `// $` delimiter. The rest of the revision entries are outside of this delimiter. StarTeam does not recognize the `$History$` keyword if you remove this line.

<code>\$Id\$</code>	Similar to <code>\$HeaderUTC\$</code> except that it is a combination of Workfile, the branch revision number (preceded by "v" for version; for example, v 1.2.1.0), UTC time, and Author. The branch revision number is in dot notation.
<code>\$Locker\$</code>	User who has the file exclusively locked (if any).
<code>\$Log\$, \$Log[x]\$, \$LogUTC\$, and \$LogUTC[x]\$</code>	File change history. <code>\$Log\$</code> is a special keyword because it expands to a multiline entry. The <code>\$Log\$</code> keyword expands to include information for each revision of the file. Revision history includes Revision Number, Date, Author, and Reason for Check In. Use <code>\$Log\$</code> to retain entries for each revision within the file. Use <code>\$Log[x]\$</code> to limit the number of revisions for which entries are retained. Replace <code>x</code> with the number of entries to be retained. For example, <code>\$Log[8]</code> saves the entries for the most recent 8 revisions. If you replace <code>x</code> with a number less than 1 or with a non-numeric character, StarTeam ignores <code>x</code> and retains all entries (as with <code>\$Log\$</code>). <code>\$LogUTC\$</code> and <code>\$LogUTC[x]\$</code> are the same as <code>\$Log\$</code> and <code>\$Log[x]\$</code> except that a UTC time replaces the local time. UTC times end in a "Z".
<code>\$NoKeywords\$</code>	Turn off keyword expansion for the rest of the file.
<code>\$Project\$</code>	Name of the project.
<code>\$Revision\$</code>	Revision number (an integer).
<code>\$Folder\$</code>	Name of the folder.
<code>\$Workfile\$</code>	Unqualified name of the working file (for example, <code>foo.cpp</code>).

Warning: Never use a keyword in a revision comment, as it will be expanded during the keyword expansion process.

Related Concepts

[Personal Options](#)

Related Procedures

[Enabling Keyword Expansion](#)

Index

- access rights
 - projects, 318
 - folder, 339
- account information, 296
- active process item, 338
- alternate applications
 - editor/merge utility/comparison utility, 280
- APEs (Alternate Property Editors)
 - projects, 320
- app-control.xml, 31
- app-control.xml file, 31
- audit log
 - filter, 599
 - email, 601
 - search, 600
- audit log entries, 602
- audit menu, 834
- branching
 - views, 169 358
 - change requests, 174
 - options, 170
 - shared items, 359
- branching shared items
 - shared items, 172
- Bulk Check-out Utility, 161
- change package object
 - View Compare/Merge, 215
- change packages
 - properties, 804 890
- Change Perspective
 - UI, 102
 - Change Package, 365 394 396 397 409 410 411 414 419 425 430 431
- change request menu, 822
- change requests
 - concepts, 61 82
 - assigning, 541
 - closing verified, 542
 - creating, 543
 - customizing filters, 550 666
 - customizing reports, 551 719 729
 - linking, 563 648
- change sessions, 217
- Chart Window
 - UI, 107
- charts
 - concepts, 90
 - chart types, 737
 - colors, 738
- check in
 - files, 507
 - comments, 518 700
- check in and out, 156
- check-in
 - atomic, 263
- check-out
 - files, 509
 - BCO, 512
- compare
 - history, 444 691
- compare non-file item properties
 - item properties, 470
- compare/merge
 - reports, 733
- connection properties, 341
- Cross-Platform client
 - UI, 97
- cross-project activities
 - projects, 131
- data
 - exporting, 725
- Datamart
 - concepts, 93
- Detail pane
 - folders, 846
 - change package, 860
 - change requests, 850
 - files, 848
 - requirement, 858
 - tasks, 853
 - topics, 856
- Detail panes
 - customize, 284
- Email notification, 841
- EOL characters, 473
- EOL Handling, 158
- exclusive locks
 - check-in, 335
- Executable Bit
 - Unix, 533
- fields
 - concepts, 92
 - customizing, 292
- File Compare/Merge
 - color of difference types, 119
 - compare, 445
 - compare/merge, 435 436 437 439 440 442 447 449 450 452 454 456 458 491 726
 - encoding, 434
 - Saving, 438 448
 - shortcuts, 777 864
 - View Compare/Merge, 120 221

- file menu, 819
- files
 - concepts, 59
 - active process items, 521 641 680
 - baselines, 683
 - check out, 514 690
 - folders not in view, 482 520
 - linking, 642 681
 - open, 528
 - project, 482 520
 - projects, 511
 - properties, 868 871 873 875 878 881
 - read-only, 525
 - rename, 496 530
 - status, 282 515 532 958 960
- filters
 - concepts, 91
 - copying, 662
 - creating, 664
 - deleting, 667
 - editing, 668
- floating items, 175
- folder menu, 816
- folder tree menu, 815
- folders
 - concepts, 56
 - items, 281 469 477 493 495 499 516 527
 - paths, 146
 - views, 344 461 468 474
- folders and files
 - visiblity, 484 522
- folders/items
 - configuring, 471
- help menu, 838
- install
 - StarTeam, 19
- items
 - detail, 478 517
 - find, 483
 - flagging, 485
 - properties, 503 536 715
- keyword expansion, 328
- keywords
 - keyword expansion, 1019
- labels, 164
 - demote, 325 621
 - folders, 463 604
 - freeze, 626
 - items, 465 606
 - promote, 334 627
 - properties, 609 707
 - revisions, 613 617 622 623 624 625 694 696 697 698
 - sorting, 630
 - views, 345 349 615 618
- language
 - local, 278
- LCOMM, 806
- linking
 - folders and items, 476 635 709
 - check in, 632
 - files, 531 649
 - items, 486 556 644
 - properties, 636 710
 - revisions, 646 701
- linking to external items
 - external links, 633
- links
 - concepts, 69 972
 - enhanced process links, 638 650 676 684
- links across servers
 - external links, 200
- location references
 - folders, 326 479
- lock/unlock
 - items, 488 523 558
- log off, 269
- log on
 - server, 271
- logon
 - options, 293
- new features, 33
- notifications
 - displaying, 294
 - open, 294
- Online Purge
 - Server, 31
- options
 - customizing, 283
- passwords, 279
- Personal Options, 151
- previous version
 - file revision, 504 537 703
- Process Items, 194
 - Change Packages, 195
- process items and process links
 - links, 197
- process rules
 - process items, 194 201
 - projects, 330 677
- process rules and items
 - active process item, 205
- process tasks
 - links, 203

- Process Tasks
 - Filter, 640 669 679
- project menu, 813
- projects, 126
 - cross-project dependencies, 129
 - enhanced process links, 321 329 639
 - names, 319
 - properties, 342
 - shortcuts, 333 337
- promotion state, 148
- promotion states, 611
- properties
 - email, 481 712
 - compare, 692 693 705 706
- queries
 - concepts, 87
 - applying, 652
 - copying, 653
 - creating, 654
 - deleting, 657
 - editing, 658
- references
 - concepts, 179
 - branch, 184
 - initial, 180
 - items, 186
 - objects, 191
 - share, 189
- Replay change package, 217
- report templates
 - customizing fields, 723 734
- reports
 - concepts, 88
 - configuring output path, 717
 - creating, 323 475 718
 - printing, 727
 - template files, 989
 - template types, 990
 - templates, 724 735
- Reports
 - templates, 989
- requirement menu, 825
- requirements
 - concepts, 66
 - creating, 575
- revisions
 - comments, 336
- selecting referenced items, 498
- Server Administration Tool
 - UI, 110 115
- servers
 - log on, 270 311
- shortcuts, 264
 - creating, 688 749
 - keyboard, 863 866
- sorting
 - grouping, 299 501 673
- task menu, 828
- tasks
 - concepts, 67
- test
 - report, 80
- Toolbar Utility
 - UI, 122
 - customizing, 285 304
 - log onto server, 270 311
 - modifying tool properties, 288 297 307 312
 - open, 313
 - refresh, 290 298 309 314
- ToolBar Utility
 - open and close, 313
- tools menu, 836
- topic menu, 831
- topics
 - concepts, 68
 - creating, 593
- URL
 - shortcut, 687 748
- VCM, 219
- VCMUtility, 209
- veresiono object properties
 - properties, 806 892
- View compare-only
 - VCM, 232
- View Compare/Merge, 113
 - automerge in VCM session, 213
 - ccommit, 371
 - change default merge types, 366
 - Compare-only, 209 259
 - compare/merge, 367 368 369 370 373 375 376 377 378 380 382 383 384 385 387 392 393 395 398 399 401 403 405 407 412 413 416 417 420 421 422 423 424 426 428 429
 - custome merge type, 389
 - default merge type options and actions, 366
 - difference colors in Compare Pane, 119 221
 - move, 256
 - Moved, 254
 - prepare data, 238
 - shares, floating, 256
 - Unchanged difference type, 254
 - VCM access rights, 233
 - VCM Change Request access rights, 235
 - VCM Child Folders access rights, 234

- VCM File access rights, 235
- VCM Floating Child Share, 251
- VCM New In Source, 253
- VCM New In Source (Promote), 253
- VCM New In Source, Pinned (Promote), 253
- VCM Parent Folder Failed, 250
- VCM Parent Folder Ignored, 250
- VCM Project access rights, 233
- VCM Requirement access rights, 236
- VCM Session Tips, 238
- VCM Target Folder Has Floating Share in Source
- View, 252
- VCM Task access rights, 236
- VCM Topic access rights, 237
- VCM View access rights, 233
- VCMUtility, 368 369 380 382 389 401 417 420
- view menu, 814
- views, 133
 - branching, 360
 - descriptions, 352
 - iterative development scenarios, 143
 - options, 1012 1015
 - properties, 354
- window menu, 837
- working folders, 154