

**AccuRev®**

## **On-Line Help Guide**

**Version 5.4  
January 2012**

*Revised 9-January-2012*

# Copyright and Trademarks

Copyright © AccuRev, Inc. 1995–2012

ALL RIGHTS RESERVED

This product incorporates technology that may be covered by one or more of the following patents:  
U.S. Patent Numbers: 7,437,722; 7,614,038.

**AccuRev**, **AgileCycle**, and **TimeSafe** are registered trademarks of AccuRev, Inc.

**AccuBridge**, **AccuReplica**, **AccuSync**, **AccuWork**, **Kando**, and **StreamBrowser** are trademarks of AccuRev, Inc.

All other trade names, trademarks, and service marks used in this document are the property of their respective owners.

# Preface

This book documents the AccuRev Java Graphical User Interface (GUI). It provides the content for the AccuRev context-sensitive help.

## Audience

This document is intended for end users and administrators who use the AccuRev Java GUI in the performance of their jobs. It is not mean to be a tutorial—we assume that you have had some form of AccuRev training.

See the AccuRev™ Installation and Release Notes for installation procedures and any late-breaking information about the current release.

## Using This Book

The contents of this book are intended to be used as on-line help within the GUI. This book format is provided only as a convenience for users who wish to browse through the help outside of the AccuRev client. Because the content was developed as on-line help, the book is not designed to be read in a linear fashion. You will find that discussions tend to jump from topic to topic with little transition material easing the way between them.

The chapters in this book are divided by funtional areas:

Chapter	Description
<a href="#">Overview of the On-Line Help</a>	Describes the AccuRev on-line help mechanism and its organization and features.
<a href="#">Using the AccuRev GUI</a>	Provides a high-level view of the AccuRev GUI.
<a href="#">The File Browser</a>	Describes in detail how to use AccuRev's features to work with your files.
<a href="#">The StreamBrowser</a>	Describes in detail AccuRev's unique stream architecture for organizing your work.
<a href="#">The History Browser</a>	Describes how to use AccuRev to view the past operations that have been performed on your files.
<a href="#">Diff, Merge, and Change Palette</a>	Explains how to view the differences between versions of your files, how to merge those differences, and how to use the AccuRev's unique Change Palette to cross-promote elements to specific streams.
<a href="#">Security</a>	Describes features such as file locking, stream locking, access control lists (ACLs), and other ways of controlling accessing to your data.
<a href="#">AccuWork</a>	Describes AccuRev's tightly integrated issue tracking feature.

Chapter	Description
<a href="#">Miscellaneous Error Displays</a>	Explains the messages that provide help buttons for more detailed information.
<a href="#">Support and Troubleshooting</a>	Explains how to contact AccuRev Customer Support, provides discussions of some commonly-encountered issues, and also describes how to configure various web browsers to work with AccuRev on-line help.

## Typographical Conventions

This book uses the following typographical conventions:

Convention	Description
<code>blue sanserif</code>	Used for sample code or output.
<code>red monospace</code>	Used for examples.
<b>bold</b>	Used for command names, and button names in the AccuSync Web user interface
<i>light italic</i>	Used for emphasis, book titles, and for first use of important terms
<a href="#">blue italic</a>	Identifies a hyperlink (to a page or Web URL, for example)

## Contacting Technical Support

AccuRev offers a variety of options to meet your technical support needs. For complete information about AccuRev technical support services, visit our Web site:

<http://www.accurev.com/support.html>

### License Issues

If you are having trouble with a license for your AccuRev product, visit the License Self Help page first to see if there is a solution for your problem:

<http://www.accurev.com/support/license-self-help.html>

### Other Support Issues

To obtain technical support for an AccuRev product:

- Go to <http://support.accurev.com/>
- Or write AccuRev technical support at [support@accurev.com](mailto:support@accurev.com)

When you contact AccuRev technical support, please include the following information:

- The AccuRev version
- The operating system
- If you are using an AccuBridge™ product:

- The AccuBridge version
- The version of the system used by the AccuBridge (JIRA or Rally, for example)
- A brief description of the problem you are experiencing. Be sure to include which AccuRev interface you were using (Web user interface, Java GUI, or CLI), any error messages you received, what you were doing when the error occurred, whether the problem is reproducible, and so on.
- A description of any attempts you have made to resolve the issue, including using the resources described in [Other Resources](#).
- A simple assessment of how the issue affects your organization.

## Other Resources

In addition to contacting AccuRev technical support, consider taking advantage of the following resources:

- AccuRev Known Problems and Solutions – <http://www.accurev.com/support/kps.html>  
The AccuRev Known Problems and Solutions page describes known problems in released versions of AccuRev products.
- AccuRev User Forum – <http://www.accurev.com/ubbthreads/>  
The AccuRev User Forum is a valuable resource. In addition to access to AccuRev's international user community, the User Forum is the place to go for resources like:
  - The AccuRev knowledge base
  - The AccuRev FAQ
 Register as a User Forum member today.
- AccuRev documentation – <http://www.accurev.com/documentation.html>  
This page provides access to AccuRev product documentation for all current and previous releases of most AccuRev products.

# Table of Contents

<b>Preface .....</b>	<b>iii</b>
Audience .....	iii
Using This Book .....	iii
Typographical Conventions .....	iv
Contacting Technical Support .....	iv
License Issues .....	iv
Other Support Issues .....	iv
Other Resources .....	v
 <b>1. Overview of the On-Line Help .....</b>	 <b>1</b>
If You Are a New AccuRev User .....	1
If You Are Upgrading From a Previous AccuRev Version .....	2
What has Changed? And Why? .....	2
What About the PDFs? .....	3
How Do I Access the Help Without Starting the Client? .....	3
How Do I Configure My Browser to Work With the New Help? .....	4
 <b>2. Using the AccuRev GUI .....</b>	 <b>5</b>
Mechanics of the AccuRev GUI .....	5
Starting the GUI .....	5
The Multiple-Tab Display .....	5
Current User, Depot, and Workspace .....	7
Keyboard Accelerators .....	7
Integration with the AccuRev Web UI .....	7
Find, Search, and Filter Features .....	9
Working with Tables .....	9
Adjusting the Widths and Order of Columns .....	10
Sorting the Rows of a Table .....	10
Using the Keyboard to Navigate through a Table .....	12
Choosing a Depot to be Used for a Command .....	12
Using the Choose Depot Dialog .....	12
Choosing a Stream to be Used for a Command .....	13
Using the Choose Stream Dialog .....	13
The File Chooser Dialog .....	13
AccuWork Export File Types .....	14
User-Specified Names for AccuRev Entities .....	14
The GUI Main Toolbar .....	15
The GUI Main Menu .....	16
The File Menu .....	16
The New Depot Command .....	16
Invoking the New Depot Command .....	16
Using the New Depot Dialog .....	16
The New Workspace Command .....	18
The New Workspace Command: Screen 2 of 3—Choosing a Name and Location for the Work-	

space) .....	20
The New Workspace Command: Screen 3 of 3—Configuring the Workspace).....	22
The New Stream and Change Stream Commands (Stream Configuration dialog) .....	23
Invoking the Command .....	23
Using the New Stream / Change Stream Dialog .....	23
The Workspaces Tab / The Open Workspace Dialog.....	25
Opening a Workspaces Tab.....	25
Invoking the Open Workspaces Command .....	26
Workspaces Tab Layout.....	26
Working in a Workspaces Tab.....	27
The New Snapshot Command .....	28
Invoking the New Snapshot Command .....	28
Using the New Snapshot Dialog.....	28
Updating a Workspace.....	28
Kinds of Changes Involved in an Update .....	29
How Update Works .....	29
When Update does not Work .....	31
More on Update .....	32
The Properties Command .....	32
Invoking the Properties Command .....	32
The Properties Display .....	32
The Issues Menu.....	33
The Tools Menu .....	33
The Synchronize Time Command .....	33
The Show Info Dialog.....	33
The Server Tasks Tab.....	34
Opening a Server Tasks Tab .....	34
Server Tasks Tab Layout.....	34
Working in the Server Tasks Tab .....	34
The Login Command.....	35
Invoking the Login Command.....	35
Using the Login Dialog.....	35
The "session" File .....	36
The Change Password Command .....	37
Invoking the Change Password Command .....	37
Using the Change Password Dialog.....	37
Filter Streams	
(Tools > Filter > Streams Command).....	37
Invoking the Filter Streams Command .....	37
About the Stream Filter.....	38
Using the Configure Stream Filter Dialog .....	38
Adding Streams .....	39
Searching by Stream Name .....	40
Filter Users/Groups	
(Tools > Filter > Users/Groups Command) .....	41
Invoking the Filter Users/Groups Command.....	41
Using the Dialog.....	41
Scope of the Users/Groups Filter .....	41
AccuRev Preferences	

(Tools > Preferences Command) .....	42
General Page .....	42
Diff/Merge Page.....	44
StreamBrowser Page .....	47
The Admin Menu .....	47
The Depots Tab .....	47
Depots Tab Layout .....	48
Working in a Depots Tab .....	48
The Reference Trees Tab.....	49
Opening a Reference Trees Tab .....	49
Reference Trees Tab Layout.....	49
Working in a Reference Trees Tab.....	50
The New Reference Tree Command.....	51
Invoking the New Reference Tree Command.....	51
Using the New Workspace Wizard.....	51
The Triggers Tab .....	51
Opening a Triggers Tab.....	51
Triggers Tab Layout.....	51
Working in a Triggers Tab.....	51
The Slices Tab .....	51
Opening a Slices Tab.....	52
Slices Tab Layout .....	52
Working in a Slices Tab .....	52
The Locks Command .....	53
Invoking the Locks Command .....	53
The Locks Dialog Box .....	53
<b>3. The File Browser.....</b>	<b>55</b>
Opening a File Browser Tab .....	55
File Browser Tab Layout.....	55
Working in the File Browser's Panes.....	57
Alternatives to the File Browser.....	57
File Browser: Browsable Data Structures .....	58
File Browser: Working in the Folders Pane.....	58
Working in Include/Exclude Mode .....	59
Folders Pane Command Reference .....	59
File Browser: Working in the Searches Pane.....	59
Performing Searches of the Entire Workspace or Stream .....	60
Search Optimizations .....	64
Deep Overlap Optimization .....	64
File Browser: Searching by Element Name .....	65
Filtering Your Search Results .....	65
File Browser: Working in the Details Pane .....	66
Details Pane Layout.....	66
Common Usage Scenarios.....	70
Details Pane Command Reference .....	79
File Browser: Controlling the Display of Element Names.....	88
AccuRev Element Status.....	88
File Status Indicators.....	89

File Status Diagram .....	91
Pathname Optimization:	
Selective Processing of External Objects .....	94
Eligible Searches .....	94
Values for ACCUREV_IGNORE_ELEMS .....	94
Timestamp Optimization:	
Controlling the Determination of (modified) Status.....	96
Traditional, Server-Based Timestamp Optimization .....	97
Local, Client-Based Timestamp Optimization.....	99
Setting and Overriding Default TSO Behavior .....	100
File Browser: Controlling the Display of External Objects .....	101
Using the File Browser's Include/Exclude Mode .....	101
File Browser Layout in Include/Exclude Mode .....	102
Working in the Rules Pane.....	103
Include/Exclude Example 1: Excluding a Directory .....	107
Include/Exclude Example 2: Simulating a Sparse Workspace .....	108
The Add to Depot Command.....	108
Invoking the Add to Depot Command.....	108
Using the Add to Depot Dialog .....	109
Results of an Add to Depot Operation -- the Details .....	109
The Anchor Command .....	110
Invoking the Anchor Command .....	110
Using the Anchor Dialog .....	111
The Annotate Tab.....	111
Opening an Annotate Tab .....	111
Annotate Tab Layout.....	111
Working in an Annotate Tab.....	112
Using the Version Timeline .....	112
The Defunct Command .....	113
Invoking the Defunct Command .....	113
Defuncting a Directory.....	114
Promoting a Defunct Element .....	114
Bringing Back a Defunct Element.....	114
The Past and Future of a Defunct Element.....	114
The Delete Command.....	115
Invoking the Delete Command.....	115
The Keep Command.....	115
Invoking the Keep Command.....	116
The Keep Dialog.....	116
Results of a Keep Operation -- the Details .....	116
The New Folder Command .....	117
Invoking the New Folder Command .....	117
Using the New Folder Dialog.....	118
The New File Command.....	118
Invoking the New File Command.....	118
Using the New File Dialog.....	118
The Patch From Command.....	119
Patch vs. Merge .....	119
Invoking the Patch From Command.....	121
The Patch Algorithm.....	121

The Populate Command .....	121
Invoking the Populate Command .....	121
Using the Populate Dialog.....	121
The Populate Progress Box.....	122
The Promote Command .....	122
Real versions and virtual versions .....	122
Inheritance of promoted versions.....	122
Overlap status and merging .....	122
Underlap status.....	122
Invoking the Promote Command .....	123
Using the Promote Dialog.....	123
The Rename Dialog.....	124
Invoking the Rename Command .....	124
Using the Rename Dialog .....	125
Renaming a Modified File Before Keeping It .....	125
Caution on Reusing the Name of a Renamed Element .....	125
If you want to return to using the original element .....	125
If you want to discard the original element and use the new element.....	126
The Revert Command.....	126
Invoking the Revert Command.....	126
Choosing a Workspace.....	127
Choosing a Workspace for the Revert Command .....	127
Revert to Backed Dialog .....	127
Trigger Firing .....	127
Revert to Most Recent Version Dialog.....	128
The Send to Issue Command .....	128
Invoking the Send to Issue Command.....	128
Specifying the Basis Version of the Change .....	128
Using the Send to Issue Dialog .....	129
Failure to Modify an Existing Change Package Entry.....	130
The Send to Workspace Command .....	131
Invoking the Send to Workspace Command .....	131
Using the Send to Workspace Dialog.....	132
The Update Progress Box.....	132
The WIP (Work in Progress) Tab .....	133
Opening a WIP Tab .....	133
WIP Tab Layout.....	133
Working in a WIP Tab.....	134
The Default Group of a Workspace or Stream .....	134
How Elements Become Members of the Default Group .....	134
How Elements are Removed from the Default Group .....	135
Viewing the Contents of the Default Group .....	135
Active Versions vs. Pending Changes in a Workspace .....	135
AccuRev Element Types .....	136
Real Versions and Virtual Versions.....	136
Real Versions.....	137
Virtual Versions .....	137

## **4. The StreamBrowser ..... 139**

Opening a StreamBrowser Tab .....	139
StreamBrowser Tab Layout .....	139
Graphical StreamBrowser Display .....	140
Tabular StreamBrowser Display .....	141
Working in the StreamBrowser .....	142
Controlling the Display .....	142
Commands Available in a StreamBrowser Tab .....	147
The Overlaps Tab .....	150
Opening an Overlaps Tab .....	150
Overlaps Tab Layout .....	150
Working in an Overlaps Tab .....	150
The Remove Stream Command	
(and the Remove Workspace Command) .....	151
Invoking the Remove Stream Command .....	151
Invoking the Remove Workspace Command .....	151
Default Group Subwindows in the StreamBrowser .....	151
Opening a Default Group Subwindow .....	151
Default Group Subwindow Layout .....	152
StreamBrowser History .....	152
Find Stream .....	153
Invoking the Find Stream Dialog .....	153
Using the Find Dialog .....	153
The Stream Diff (Files) Tab	
(Show Diff By Files command) .....	154
Opening a Stream Diff (Files) Tab .....	154
Stream Diff (Files) Tab Layout .....	154
Working in a Stream Diff (Files) Tab .....	156
The Stream Issues and Stream Diff (Issues) Tabs .....	157
Opening the Tab .....	157
Stream Issues Tab Layout .....	157
Working in the Issues Pane .....	158
Working in the Change Package Contents Pane .....	162
The Issue Dependencies Tab .....	163
Opening an Issue Dependencies Tab .....	163
Change Package Dependencies .....	164
Issue Dependencies Tab Layout .....	165
Working in the Issues Pane .....	167
Working in the Change Package Contents Pane .....	168
The Active Transactions Tab	
(Show Active Transactions command) .....	168
Opening an Active Transactions Tab .....	169
Active Transactions Tab Layout .....	169
Working in a Active Transactions Tab .....	169
The Stream Version Browser .....	170
Opening a Stream Version Browser Tab .....	170
Stream Version Browser Tab Layout .....	170
Working in a Stream Version Browser Tab .....	171
Patches and Change Packages .....	172
Structure of a Patch .....	172
Patch Display .....	174

Structure of a Change Package .....	174
Change Package Display .....	176
Creating a New Change Package Entry .....	176
Modifying Existing Change Package Entries .....	176
Change Package History .....	179
Change package -- part of an issue record, or just associated with it? .....	179
Unaffiliated Changes ("Dark Matter") .....	180
The Patch List Tab	
(Show Patch List command) .....	182
Opening a Patch List Tab .....	183
Patch List Tab Layout .....	183
Working in a Patch List Tab .....	183
Streams and Issues .....	184
Change Packages "In" Streams .....	184
<b>5. The History Browser .....</b>	<b>187</b>
Opening a History Browser Tab .....	187
History Browser Tab Layout .....	188
The Summary Pane .....	188
The Comment Pane .....	189
The Versions Pane .....	189
Working in the Summary Pane .....	190
Summary Pane Controls .....	190
Working in the Versions Pane .....	192
History Browser / Summary Pane Controls	
Go to a Particular Transaction .....	194
History Browser / Summary Pane Controls	
Display Transactions with a Particular Comment .....	194
History Browser / Summary Pane Controls	
Manage Groups of Transactions .....	194
History Browser / Summary Pane Controls	
Display Transactions Created in a Particular Interval .....	195
History Browser / Summary Pane Controls	
Display a Particular User's Transactions .....	196
The Version Browser .....	196
Opening a Version Browser Tab .....	196
Version Browser Tab Layout .....	197
Ancestry Lines .....	197
Ancestry Relationships .....	198
Closest Common Ancestor .....	202
Operations on Versions .....	203
Version Browser Preferences .....	205
<b>6. Diff, Merge, and Change Palette .....</b>	<b>207</b>
The Diff Tool .....	207
Invoking the Diff Tool (or Another File-Comparison Tool) .....	207
Diff Tab Layout .....	208
Difference Section Color-Coding .....	209
Navigating in a Diff Tab .....	210

Diff Toolbar Navigation Buttons .....	210
Difference Maps .....	212
Editing a File Using the Diff Tool .....	212
Diff Toolbar Edit-by-Diff Buttons .....	214
Main Menu Commands Available in a Diff Tab .....	214
The Change Palette Tool.....	215
Opening a Change Palette Tab .....	215
Change Palette Tab Layout .....	220
Layout of the Specified Versions Pane .....	220
Layout of the Merged Versions Pane .....	222
Working in a Change Palette Tab .....	223
Specifying the Destination Stream .....	223
Sending Versions to the Destination Stream .....	224
Merging the Source and Destination Versions .....	225
Selecting a Workspace for Performing Merges .....	226
Patching Changes from the Source Version into the Destination Version .....	227
Multiple Entries for the Same Element .....	227
Commands Available in a Change Palette Tab.....	227
The Merge Tool.....	229
The Merge Command.....	229
What about elements that aren't text files?.....	229
Configuring the GUI to use another text-file-merge tool .....	229
Why Do I Need to Merge? .....	229
Invoking the Merge Command.....	230
The AccuRev Merge Tool.....	231
Merge Tab Layout .....	231
Resolving Conflicts .....	233
Commands Available in a Merge Tab.....	239
The Search Command .....	241
Invoking the Search Command .....	241
Using the Search Dialog.....	241
The Merge, Patch, and Reverse Patch Algorithms.....	242
Merge: Incorporating Content Changes .....	243
Patch: Incorporating Content Changes .....	245
Reverse Patch: Removing Content Changes.....	246
Resolving Namespace Conflicts.....	247
Kinds of Namespace Changes .....	247
Resolving a "Rename" Conflict.....	248
Resolving a "Move" Conflict .....	248
Merging Versions of a Binary File .....	249
Using the Binary Merge Dialog .....	249
Merging Versions of an HTML File.....	250
Merging Versions of a Link .....	250
Basic Merge Algorithm .....	250
What Constitutes a Change to a Link? .....	250
Link Merge Procedure .....	251
Merging Versions of a Directory.....	251
Basic Merge Algorithm .....	251
What Constitutes a Change to a Directory?.....	252
Directory Merge Procedure .....	252

The Merge Complete Dialog .....	252
Using the Merge Complete Dialog .....	252
<b>7. Security .....</b>	<b>255</b>
Users and Groups .....	255
User Authentication .....	255
Locks on Streams .....	256
Access Control List (ACL) Permissions .....	256
Restricting Access to Commands using Triggers .....	257
Which Security Feature Should I Use? .....	258
File Locking .....	259
Serial Development and Parallel Development .....	259
The Limited Effect of an Exclusive File Lock .....	260
Anchor-Required Workspaces .....	260
The Security/Users Subtab .....	261
Opening a Security/Users Subtab .....	261
Security/Users Subtab Layout .....	261
Working in a Security/Users Subtab .....	262
Controlling Which Users are Displayed .....	262
Operating on a Selected User .....	262
The Security/Groups Subtab .....	263
Opening a Security/Groups Subtab .....	263
Security/Groups Subtab Layout .....	263
Working in a Security/Groups Subtab .....	264
Controlling Which Groups are Displayed .....	264
Creating a New Group .....	264
The Security/ACL Subtab .....	265
Opening a Security/ACL Subtab .....	265
Security/ACL Subtab Layout .....	265
How Individual Commands Use the ACL .....	266
Conflicting Permissions .....	267
Working in a Security/Access-Control Subtab .....	268
The Available AccuRev Servers Dialog (and the New Server and Edit Server dialogs) .....	269
Invoking the Available AccuRev Servers Dialog .....	269
Using the Available AccuRev Servers Dialog .....	269
<b>A. Miscellaneous Error Displays .....</b>	<b>271</b>
No Query Results Error .....	271
Diff Previous Transaction Error .....	271
The Issues Conflict Message Box .....	271
No Schema Error .....	271
Promote Coalesce Error .....	271
Which update topic? .....	272
<b>B. AccuWork .....</b>	<b>273</b>
Working with Issue Records .....	273
The Edit Form Tab .....	273
Opening an Edit Form Tab .....	273

Edit Form Tab Layout .....	274
Working in an Edit Form Tab .....	274
The Edit Form Tab -- "Changes" Subtab .....	282
"Changes" Subtab Layout.....	282
Working in the "Changes" Subtab.....	283
The Edit Form Tab -- "Issue History" Subtab .....	284
"Issue History" Subtab Layout.....	284
Working in an "Issue History" Subtab.....	285
The New File Attachment Dialog.....	285
Invoking the New File Attachment Command.....	285
Using the Browse for Attachment Dialog.....	285
The New URL Attachment Dialog.....	286
Invoking the New URL Attachment Command.....	286
Using the New URL Dialog.....	287
Issue Record Queries .....	287
The AccuWork Queries Tab.....	287
Opening a Queries Tab .....	287
Query Mode Layout .....	288
Working in Query Mode .....	290
The AccuWork Query Editor .....	296
Opening the Query Editor .....	296
Query Editor Layout.....	296
Working in the Query Editor.....	297
The Setup Columns Command .....	301
Invoking the Setup Columns Command .....	301
Using the Setup Columns Dialog.....	302
Schema Editor .....	303
The AccuWork Schema Editor -- Overview .....	303
Invoking the Schema Editor .....	303
Saving Changes to the Schema .....	303
The AccuWork Schema Editor (Schema subtab) .....	304
Adding and Removing Fields from the Schema.....	305
The AccuWork Schema Editor (Layout subtab) .....	311
Contents of the Layout Subtab .....	312
Form Layout Operations.....	313
The AccuWork Schema Editor (Lists subtab) .....	316
Lists Subtab Layout .....	316
Lists Subtab Operations .....	316
The AccuWork Schema Editor (Relationship Types subtab) .....	317
Relationship Types Subtab Layout .....	318
Working on a Relationship Types Subtab .....	319
The AccuWork Schema Editor (Validation subtab).....	319
Initializing Field Values in a New Issue Record .....	320
Conditional Validations .....	321
Revising the Choices for a "choose" Field .....	323

The AccuWork Schema Editor (Change Packages subtab) .....	325
Change Packages Subtab Layout .....	325
Working in the Change Package Results Section .....	327
Working in the Change Package Triggers Section .....	327
Change-Package-Level Integration between AccuRev and AccuWork .....	328
Enabling the Integration .....	328
Triggering the Integration .....	330
What if Both Integrations are Enabled? .....	331
Transaction-Level Integration between AccuRev and AccuWork .....	331
Enabling the Integration .....	331
Triggering the Integration .....	332
Implementation and Customization of the Transaction-Level Integration .....	332
If Both Integrations are Enabled .....	333
<b>C. Support and Troubleshooting .....</b>	<b>335</b>
Customer Support Notes .....	335
Metadata collection .....	335
Database inconsistencies .....	336
Running the AccuRev GUI in debug mode .....	336
User authentication problems .....	337
Getting AccuRev Server and client version information .....	337
Integration issues .....	338
AccuRev license issues .....	338
Potential AccuRev defects .....	338
Replication problems .....	339
Performance problems .....	339
Server task list .....	339
If you are not seeing an element .....	340
The Logs Tab .....	340
Opening a Logs Tab .....	341
Logs Tab Layout .....	341
Working in a Logs Tab .....	341
Configuring a Web Browser to View Help Topics .....	342
Notes on Internet Explorer .....	343
Notes on Google Chrome .....	343
<b>Index .....</b>	<b>345</b>


# 1. Overview of the On-Line Help

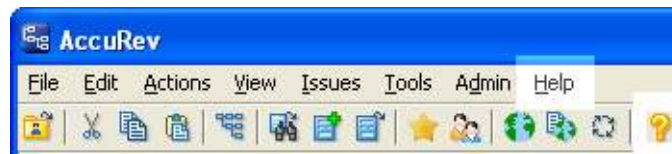
This document contains all of the information available in the AccuRev Java Client on-line help. When you click on a Help button in AccuRev, you will be taken to a page somewhere within this book.

This chapter provides a quick overview of the AccuRev Help mechanism. The help system was upgraded in AccuRev Release 5.2, so there is information here of interest for users who are familiar with the older help system found in releases prior to 5.2.

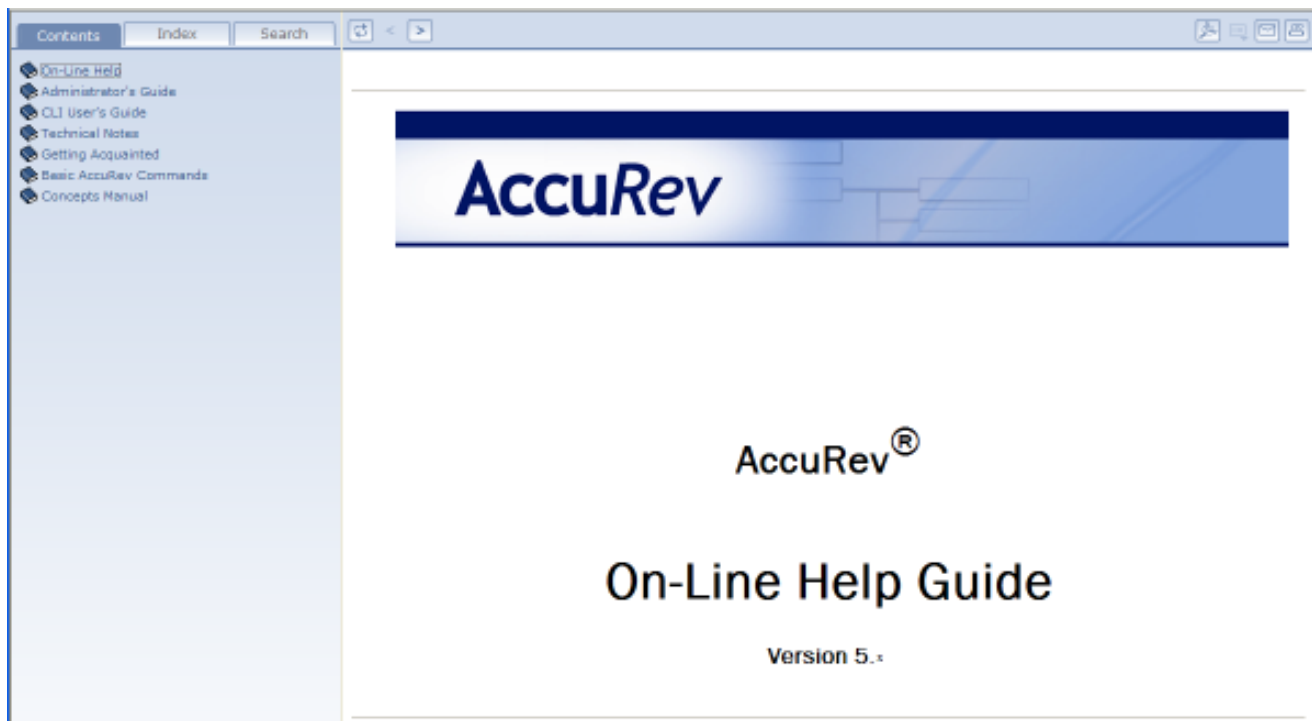
## If You Are a New AccuRev User


The AccuRev Java client provides extensive documentation in the form of a context-sensitive help system. This help not only covers the Java Graphical User Interface (GUI), but also provides access to the complete AccuRev documentation set, in both HTML and PDF format. The help can be invoked

from AccuRev with the Help menu along the toolbar, or from any Help  button.




If you click on Help Contents & Manuals, you will see a browser that includes a navigation pane, containing Contents (TOC), Index, and Search tabs.



If you click on a Help button, or click on the Help menu option for “This Screen”, you will see a smaller browser display focused on the topic of interest. The navigation pane is collapsed to save screen space, but it can be displayed by clicking on the Show Navigation  button in the upper-left corner of the display.



Note: Once the Navigation pane has been opened, it cannot be closed, but you can click and drag its right border toward the left and make it almost zero width.

All documentation is provided in both HTML and PDF. Typically you search or browse to the information you need in the HTML. If you decide you want to view the PDF version of the current document, you click the PDF button  in the upper-right of the Help window.

## If You Are Upgrading From a Previous AccuRev Version

If you have used the AccuRev Java client on-line help in releases previous to 5.2, you will notice a number of differences. If you have used recent versions of the AccuRev Web UI, or the “manuals” help that was introduced AccuRev Releases 4.9 and 5.1, you might recognize some of the features, and notice that all AccuRev help has now been standardized on a single platform.

### What has Changed? And Why?


The new on-line help allows us to respond to several customer requests:

- We have brought back a PDF version of the GUI Help documentation, which was removed from the doc set several releases ago. This PDF, named the AccuRev On-Line Help Guide, is now a standard part of the AccuRev documentation set, just like the Administrator's Guide or the CLI User's Guide.
- The entire document set (except for AccuBridge, WebUI Help, and Release Note documents) are now packaged together with the on-line help, and can be searched and viewed with a single mechanism. The Contents and Index from all documents are now available from tabs displayed in the help window navigation pane. If you want to find all references to "Change Packages", you can enter this query in the Search tab, and find all references to this term across all six of the books in the documentation set. (And the results are weighted with a "relevance score", so you avoid wasting time looking at relatively minor references). If you use a browser that accepts cookies from a local file system, you will also see a Favorites tab in the navigation page, which allows you to store links to frequently-accessed topics.
- We can now provide cross-links between the on-line help and the doc set, so if a help discussion refers to a chapter in the Administrator's Guide, you can click on a link and be taken directly to that discussion.

Since the new Help system is now based on the same document source as the rest of the documentation set, it is easier for us to share content between documents, and to consolidate redundant sections. Expect the documentation to undergo constant reorganization and improvement in future releases.

## What About the PDFs?

The new help system provides both HTML and PDF versions of all documents. Whenever you have an AccuRev help window open, you can access the PDF for the currently displayed document by

clicking the PDF button  in the upper-right of the Help window.

To achieve this integration, the help system installs its PDFs in document-specific subfolders on the client machine, such as `<accurev_install>/doc/webHelp/AccuRev_Admin/AccuRev_Admin.pdf`, `<accurev_install>/doc/manual_help/AccuRev_User_CLI/AccuRev_User_CLI.pdf`, etc.

However, long time users of AccuRev are accustomed to finding all AccuRev documentation PDFs at `<accurev_install>/doc`. We heard from Release 4.9 users that the new, nested folders were too hard to access, so we now also provide a copy of the PDFs (including the new On-Line Help Guide) in the traditional `<accurev_install>/doc` directory.

Also, Windows users who are accustomed to having direct access to the PDFs through the Start -> AccuRev menu will find that this avenue is still available.

## How Do I Access the Help Without Starting the Client?

Many AccuRev users like to access the documentation without necessarily starting the AccuRev client. This is easy: make a new bookmark in your browser to `<accurev_install>/doc/webHelp/AccuRevHelp.htm`

## How Do I Configure My Browser to Work With the New Help?

As browsers have matured, they have often added tighter security requirements with new releases. Since the AccuRev help is based on Javascript, and is accessed from a local file system rather than through a web server, you may need to configure the security settings on your preferred browser to display the help. If the AccuRev help fails to display properly the first time you try to use it, please see [Configuring a Web Browser to View Help Topics](#) in [Appendix C Support and Troubleshooting](#) for more details, and check the Release Notes for any late-breaking changes.

## 2. Using the AccuRev GUI


This chapter describes the basics of using the AccuRev Java GUI.

### Mechanics of the AccuRev GUI

The following sections discuss overall aspects of the AccuRev GUI display and user interface.

#### Starting the GUI

You can start the AccuRev GUI from the desktop (Windows only) or from a command shell:

- On the Windows desktop, double-click the  AccuRev icon. There is also an AccuRev group in the Start > Programs menu.
- In a command shell, use the `acgui` command:

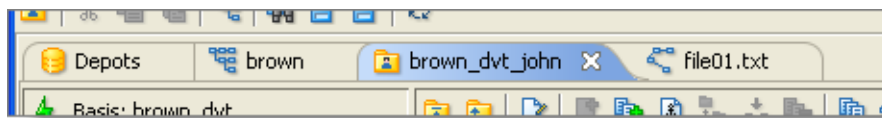
```
> acgui      (Windows)
> acgui &    (UNIX)
```

The way the GUI window initially appears varies. It might be blank, or it might open one or more tabs, displaying your work from the preceding GUI session. The AccuRev GUI reopens:

- [The File Browser](#) tabs, each displaying the contents of an individual workspace or stream.
- [The StreamBrowser](#) tabs, each displaying the hierarchy of streams, snapshots, and workspaces for a particular depot.
- The [AccuWork](#) tabs, each showing the set of AccuWork queries defined for a particular depot (and, if you've defined a **default query**, its results).

#### The Multiple-Tab Display

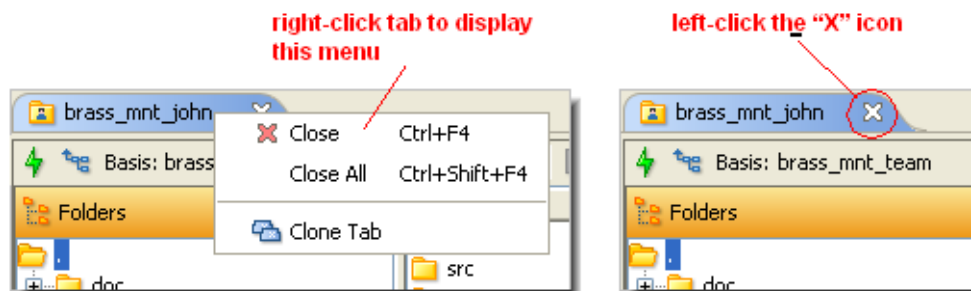
The AccuRev GUI window uses multiple tabs to enable you to switch quickly among several activities. For example, you might wish to switch among:



- Viewing the contents of your main workspace, using the File Browser.
- Viewing some or all the versions of a particular file, using the Version Browser.
- Viewing a list of all **depots** in the AccuRev **repository**.
- Viewing the streams in a particular depot, using the StreamBrowser.

## Closing Tabs

At any time, you can "clean up", by closing one or more of the tabs, in either of these ways:

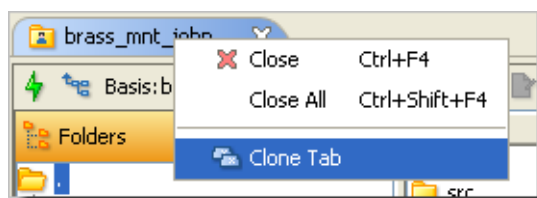


- Right-click on a tab control to display its context menu, then select Close.
- Left click the "X" icon on the tab control.

## Cloning Tabs

The AccuRev GUI assumes that you want to avoid having multiple tabs with the same contents. So, for example, if there's already a File Browser tab on workspace *brass\_mnt\_john* and you execute an *Open Workspace* command on the same workspace, the GUI simply brings the existing tab to the front.

If you do want to have two or more tabs showing the same data structure, use the File > Clone Tab command.

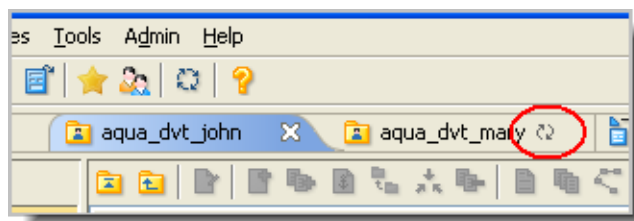


This enables you, for example, to work in two different parts of a large stream hierarchy (two StreamBrowser tabs), or to work in two different folders of a workspace (two File Browser tabs).

Note: The History Browser, Version Browser, and Stream Version Browser do not support the tab-cloning scheme.

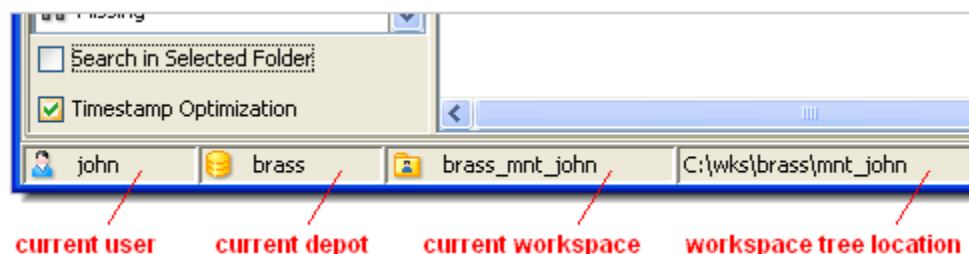
## Refreshing Tabs

The data displayed on a GUI tab can become out of date as a result of your work on other GUI tabs, your work using the AccuRev CLI, and/or other users' work. AccuRev displays a "refresh" icon next to the tab's data if it determines that the tab's data may be out of date.



The command *View > Refresh* (or function key **F5**, or the main GUI menu toolbar *Refresh* icon) updates a tab's data. You can configure the GUI to refresh tab data automatically whenever you switch tabs: invoke the command *Tools > Preferences* , and clear the *Require Manual Refresh* checkbox.

## Current User, Depot, and Workspace



With a few exceptions, you must be logged in (See [“The Login Command”](#) on page 35.) as a particular AccuRev *user* in order to execute AccuRev commands. An indicator in the lower left corner of the GUI window shows the **current user** (the one who is currently logged in).

Typically, the AccuRev repository is organized into multiple *depots*. For example, there might be separate depots for the development, testing, and documentation groups. Most GUI tabs display the data from one particular depot. When you're using a particular tab, the associated depot is termed the **current depot**. Its name is displayed at the bottom of the GUI window.

Many tabs have a **current workspace** context: the workspace that is the source of (or the possible destination of) the data that the tab displays. The name of the current workspace is displayed at the bottom of the GUI window, along with the pathname of its *workspace tree*.

## Keyboard Accelerators

In a dialog box, you can "press" buttons using **Alt** key sequences, such as **Alt-C** for *Cancel*. The **Esc** key also performs a *Cancel* operation in any dialog box.

In some dialog boxes, you don't even need to use the **Alt** key when invoking a keyboard accelerator: In a "Yes/No" dialog box, you can type **Y** or **N** (either uppercase or lowercase). In a message or confirmation box (no input field), you can indicate *Ok* by typing **O** (either uppercase or lowercase), or by pressing **Enter**.

Many commands can be invoked with a keyboard accelerator, without opening any menu at all. The accelerators for such commands are listed on the menu itself.

## Integration with the AccuRev Web UI

In addition to the Java GUI described in this document, AccuRev also provides a Web User Interface (Web UI) which gives users access to many non-filesystem-based AccuRev features from any supported web browser. The Web UI server must be installed and configured by an AccuRev administrator, who will provide AccuRev users with a web address through which they can access the Web UI functionality.

Once configured, the Web UI can be easily accessed from the Java client via two buttons on the main menu bar:



### Open in Web

When you click this button, the current view will be opened in a web browser.



### Copy URL to Clipboard

Stores a URL that you can use to bring up the current view in a web browser. This is useful if you would like to share the current view with another user. Use keyboard sequence CTRL-V, or an application “paste” command to retrieve the text from the clipboard. (For example, Windows users might open Notepad, and use the Edit>Paste command to enter the URL into the open document.)

Also, the Web UI will automatically come up when you access an AccuWork issue if any of the following three conditions is true:


- If the separately-licensed AccuWorkflow product is enabled.
- If the AccuRev administrator has configured the site to view issues only in the Web UI (see [Configuring the Web User Interface \(Web UI\)](#) on page 28 of the AccuRev [Administrator’s Guide](#)).
- If you have set your local preferences to view issues only in the Web UI (see [AccuRev Preferences \(Tools > Preferences Command\)](#) on page 42).

There are certain views that cannot be opened from the Java GUI in the Web UI. For example, manipulation of files in a local file system cannot be performed from a web browser for security reasons, so while you can get a File Browser web view of a dynamic stream, you cannot bring up a File Browser web view of your workspace. In these cases, you will see the message “URL not available for this view” when you try to use these buttons.

Note: For these features to function, the AccuRev Web UI server must have been installed and configured with your AccuRev server. If your web browser or clipboard show a URL containing the string “null” (for example, `null/stream/1/?view=sb`), the AccuRev Web UI server is not available. Information about installing and configuring the AccuRev Web UI server is available to AccuRev administrators in the AccuRev Installation and Release Notes, the AccuRev Web Interface Installation and Release Notes, and the AccuRev Administrator’s Guide.

While some features that are available in the Java GUI are not available in the Web UI, other features (such as the separately-licensed AccuWorkflow product, or the Bulk Update feature for AccuWork issues) are available only in the Web UI.

## AccuWorkflow

The separately-licensed AccuWorkflow product was previously available in the Java GUI. As of 5.2, this functionality is now exclusively available from the Web UI. If AccuWorkflow is installed and configured at your site, the Web UI will automatically launch whenever needed from the Java GUI. For more information about AccuWorkflow, see the AccuRev Web Interface User’s Guide, which is available from the  Help button on the Web UI toolbar.

Important: If you have AccuWorkflow enabled, and use a New Issue button from within a dialog in the Java client, you must set any AccuWorkflow fields in those issues manually.

## Find, Search, and Filter Features

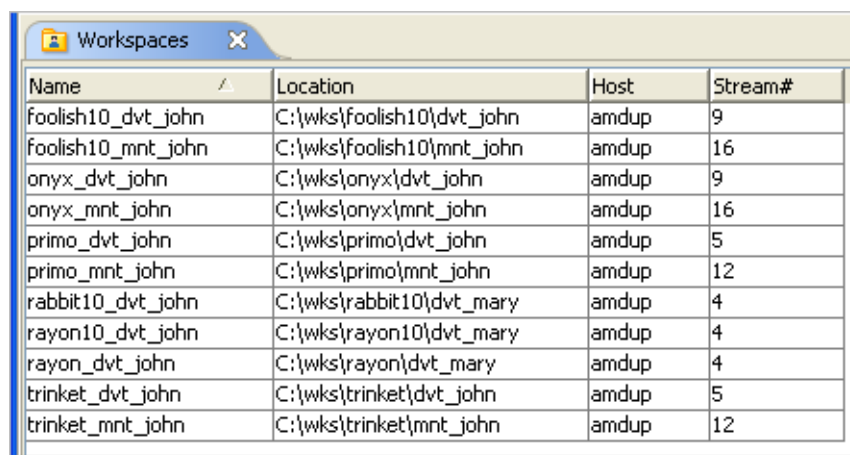
The AccuRev GUI provides several ways to locate and filter information:

- To display elements with specific statuses, see [File Browser: Working in the Searches Pane](#) on page 59
- To limit the number of streams displayed in the Streambrowser, see [Quick Stream Filtering Options](#) on page 40
- To search for elements by name in your Workspace, see [File Browser: Searching by Element Name](#) on page 65
- To find streams by name in the StreamBrowser, see [Find Stream](#) on page 153
- To filter the results of the “find streams by name” and “search for elements by name” features, see [Displaying and Filtering Stream Find Results](#) on page 154 and [Filtering Your Search Results](#) on page 66.
- To filter the results of various list dialogs, such as the Choose Depot dialog (see [Filter Results](#) on page 12).

## Working with Tables

In many contexts, the AccuRev GUI displays data in the form of a rows-and-columns table.

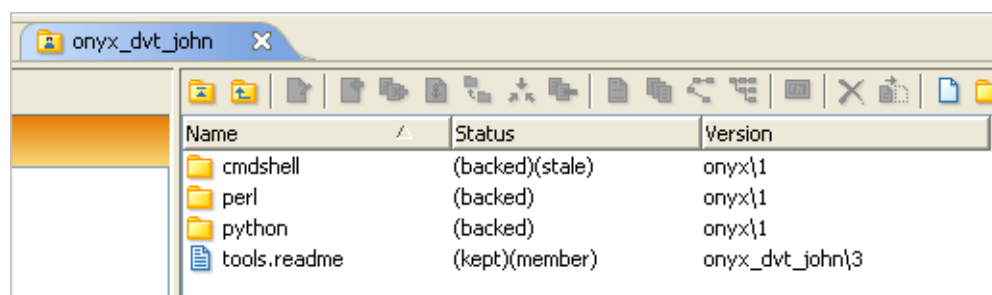
Example: File Browser's Details pane



The screenshot shows a window titled 'Workspaces' with a table containing the following data:

Name	Location	Host	Stream#
foolish10_dvt_john	C:\wks\foolish10\dvt_john	amdup	9
foolish10_mnt_john	C:\wks\foolish10\mnt_john	amdup	16
onyx_dvt_john	C:\wks\onyx\dvt_john	amdup	9
onyx_mnt_john	C:\wks\onyx\mnt_john	amdup	16
primo_dvt_john	C:\wks\primo\dvt_john	amdup	5
primo_mnt_john	C:\wks\primo\mnt_john	amdup	12
rabbit10_dvt_john	C:\wks\rabbit10\dvt_mary	amdup	4
rayon10_dvt_john	C:\wks\rayon10\dvt_mary	amdup	4
rayon_dvt_john	C:\wks\rayon\dvt_mary	amdup	4
trinket_dvt_john	C:\wks\trinket\dvt_john	amdup	5
trinket_mnt_john	C:\wks\trinket\mnt_john	amdup	12

Example: Workspaces tab



The screenshot shows a window titled 'onyx\_dvt\_john' with a table containing the following data:

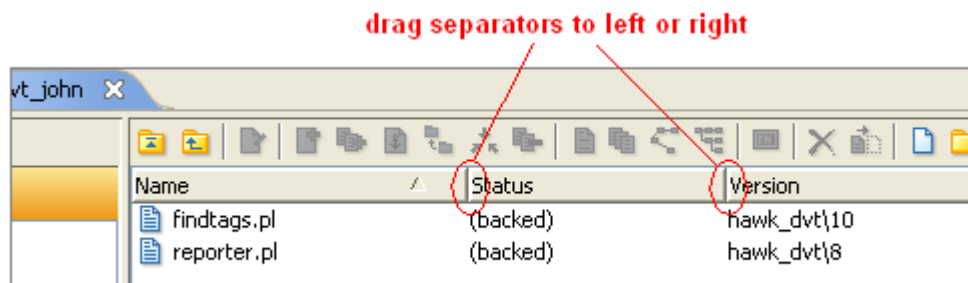
Name	Status	Version
cmdshell	(backed)(stale)	onyx\1
perl	(backed)	onyx\1
python	(backed)	onyx\1
tools.readme	(kept)(member)	onyx_dvt_john\3

The AccuRev GUI often presents information in the form of a table, with multiple rows and multiple columns. You can adjust all such tables, to maximize their usefulness, as described in the following sections.

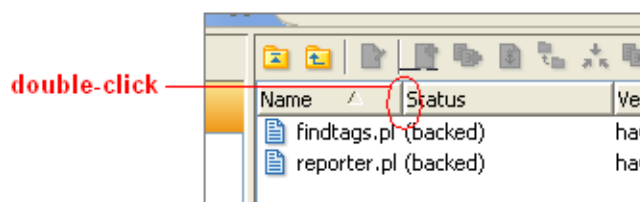
## Adjusting the Widths and Order of Columns

In any table, you can adjust columns widths and change the column order as follows:

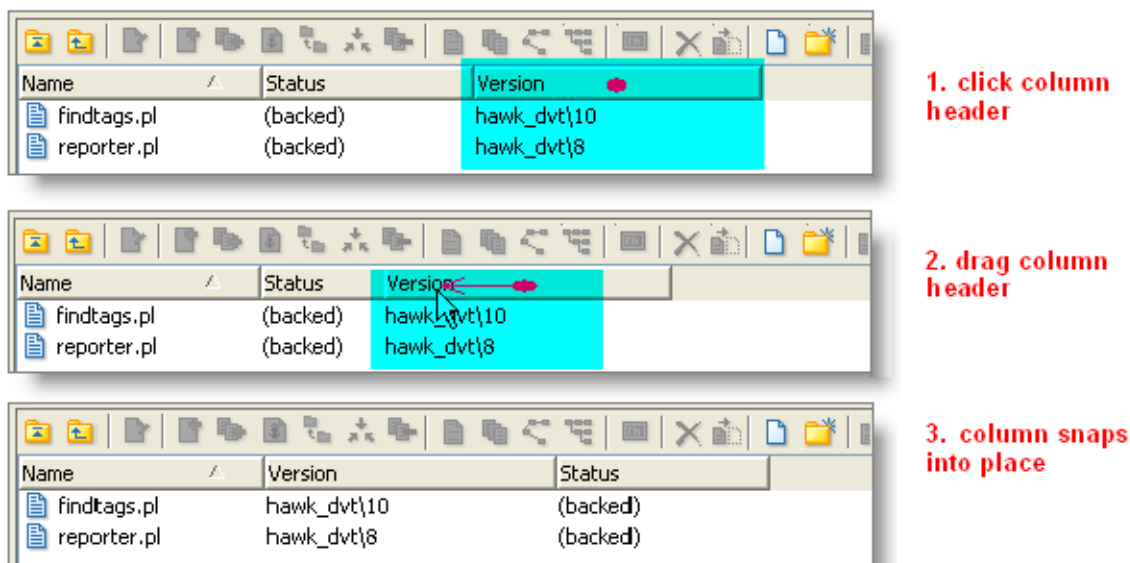
- To resize the columns, click and drag the column separators.



- Double-clicking the column separator to a column's right resizes the column to exactly fit its longest value.



- To rearrange the columns, click and drag the column headers.



## Sorting the Rows of a Table

Initially, the rows of a table are sorted on one column (**single-column mode**). A direction icon in the header for that column indicates whether the sort is lowest-to-highest or highest-to-lowest.

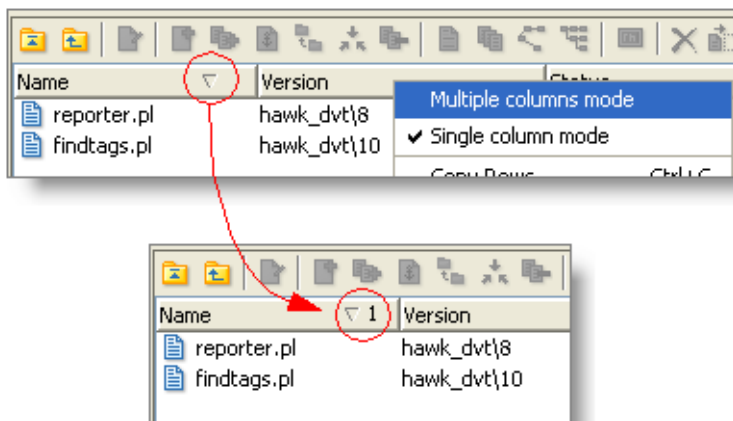


Note: In the Details pane of the File Browser (and any other table that lists file names and directory names in the same column), a lowest-to-highest sort places all directories before all files; a highest-to-lowest sort places all files before all directories.

Left-click on any column header to continue in this mode:

- Click on the current sort column to reverse the sort order.
- Click on another column to make it the sort column.

You can switch at any time to **multiple-columns mode**, in which you define a primary sort column, a secondary sort column, and so on. Right-click any column header to switch sort modes.



A "1" appears next to the direction icon in the current sort column, indicating that this is now the primary sort column.

Click another column to make it the secondary sort column. A direction icon annotated with "2" appears in this column, and the rows are reordered according to the two-level sort. Continue in this way to define additional sort levels.

In multiple columns mode, you can:

- Left-click on any column header without a direction icon: this defines an additional sort level, using that column.
- Left-click on any column header that already has a direction icon: this reverses the sort order at that sort level.
- Left-click on any sort column's level number (1, 2, etc.) to change the sort level of that column.

The table's rows are resorted automatically each time you perform any of these actions.

## Using the Keyboard to Navigate through a Table

The keyboard's navigation keys—up-arrow, down-arrow, **PgUp**, **PgDn**, **Ctrl-PgUp**, **Ctrl-PgDn**—move a table's selection highlight in the expected way. In addition, you can navigate by typing any alphanumeric character; this moves the selection highlight to the next row whose entry in the (primary) sort column begins with that character.

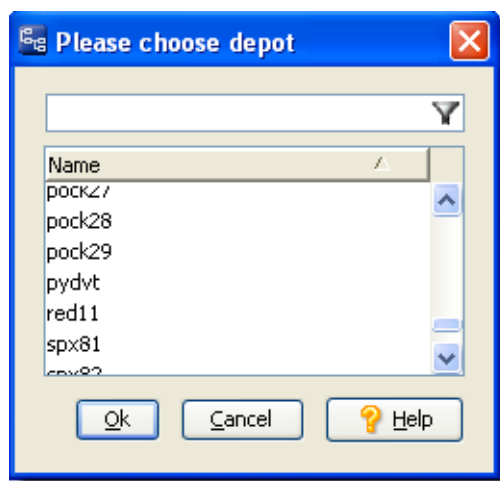
## Choosing a Depot to be Used for a Command

In many situations, AccuRev must execute a command in the context of a particular *depot*. In many cases, it selects the depot context automatically. In other cases, it prompts you to select a depot from a list of all the depots in the AccuRev *repository*.

### Using the Choose Depot Dialog

The depot names are initially sorted alphabetically (A-Z). You can click the **Name** header to reverse the sort order (Z-A). Choose a depot in either of these ways:

- Select a depot name and click Ok.
- Double-click a depot name.



### Filter Results

Optionally, use the filter field at the top of the dialog to narrow down your results. If only one element remains, AccuRev automatically selects it.

## Choosing a Stream to be Used for a Command

In some situations, AccuRev prompts you to select a stream from a list of all the streams in a particular depot. If you have set a Stream Filter (See [“Filter Streams \(Tools > Filter > Streams Command\)”](#) on page 37.), not all streams appear in this dialog.

### Using the Choose Stream Dialog

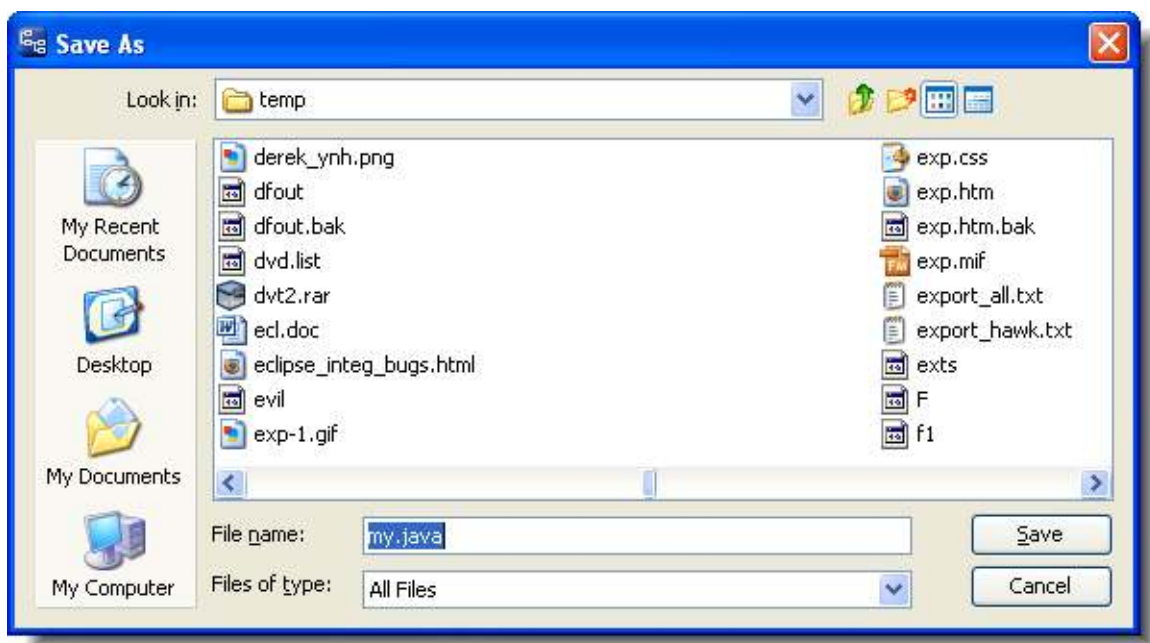
The stream names are initially sorted alphabetically (A-Z). You can click the **Name** header to reverse the sort order (Z-A).

Choose a stream in any of these ways:

- Select a stream name and click Ok.
- Double-click a stream name.


## The File Chooser Dialog

In many contexts, AccuRev prompts you to specify a location in your machine's local disk storage.



In each context where you choose a file for input or output, AccuRev suggests the directory you most recently used in that context (or defaults to your home directory).

Where appropriate — for example, in specifying attachments to an issue record — you can specify multiple files at the same time using the standard multiple-selection mechanism: hold down the CTRL key while selecting the items .

In some contexts in which you specify a file (or other data structure) to be created, the dialog includes a  *New Folder* button. This enables you to create a new directory 'on the fly' in which to create the data structure.

## AccuWork Export File Types

The AccuWork *Export* and *Export Table* commands create a file from the contents of one or more *issue records*. In these contexts, the export file must have one of these formats, which you choose from the File Type listbox:

### HTML

Standard Web-page format. On Windows machines, AccuWork automatically invokes your default web browser, to display the export file.

The contents of the issue record are rendered as one or more HTML tables. In a web browser, each table cell changes height and width when you adjust the size of the browser window. This is particularly useful for viewing the contents of multiple-line text fields.

### XML

A more highly structured document, suitable for data interchange. Many web browsers can display XML documents.

Note: The XML-format file does not include a DTD or Schema.

### CSV

("comma-separated values") A line-by-line text format, suitable for importing into various applications. On Windows machines, AccuWork automatically attempts to invoke Microsoft Excel, to display the export file.

## User-Specified Names for AccuRev Entities

AccuRev deals with many kinds of named entities: files, directories, pathnames, depots, users, and so on. The table below details the restrictions on user-defined names for the various entities. Note that all names are case-sensitive; for example, user *john* is not the same as user *John*.

**Note:** In the table below, "characters" means NON-NULL characters. Do not subtract 1 to account for the terminating NUL.

Kind	Maximum Name Length	Characters Allowed in Name	Character Restrictions
depot stream snapshot workspace reference tree	79	A-Z,a-z,0-9 - _ . + @ <i>SPACE</i>	cannot begin with a digit or with "."
user group	99	same as above	must begin with a letter
password	19	any	none (caution: this means that special characters, such as <i>BACKSPACE</i> are valid password characters)
pathname of slice (of a depot or replica)	255	name validation performed by the operating system	applied by the operating system

pathname of workspace pathname of reference tree segment (between slashes) of element pathname	127	name validation performed by the operating system	applied by the operating system
	255	name validation performed by the operating system	applied by the operating system

## The GUI Main Toolbar

The main toolbar gives you quick access to commonly used features:



From left to right, the icons are:



Open Workspace

See: [Opening a Workspaces Tab](#) on page 25



Cut, Copy, Paste

See: [Renaming or Moving a File](#) on page 72, [Changing a Directory](#) on page 75, or [Manual Editing](#) on page 237.



View Streams

See: [Opening a StreamBrowser Tab](#) on page 139



Queries, New Issue, Open Issue

See: [The AccuWork Queries Tab](#) on page 287, and [The Edit Form Tab](#) on page 273



Configure Stream Filter, Configure Users/Groups Filter

See: [Filter Streams \(Tools > Filter > Streams Command\)](#) on page 37, and [Filter Users/Groups \(Tools > Filter > Users/Groups Command\)](#) on page 41



Open in Web, Copy URL to Clipboard, Refresh

See: [Integration with the AccuRev Web UI](#) on page 7, and [Refreshing Tabs](#) on page 6



Help

See: [Overview of the On-Line Help](#) on page 1

# The GUI Main Menu

The following sections describe the menus found along the toolbar of the GUI display.

## The File Menu

## The New Depot Command

The *New Depot* command creates a new *depot* in the AccuRev *repository*.

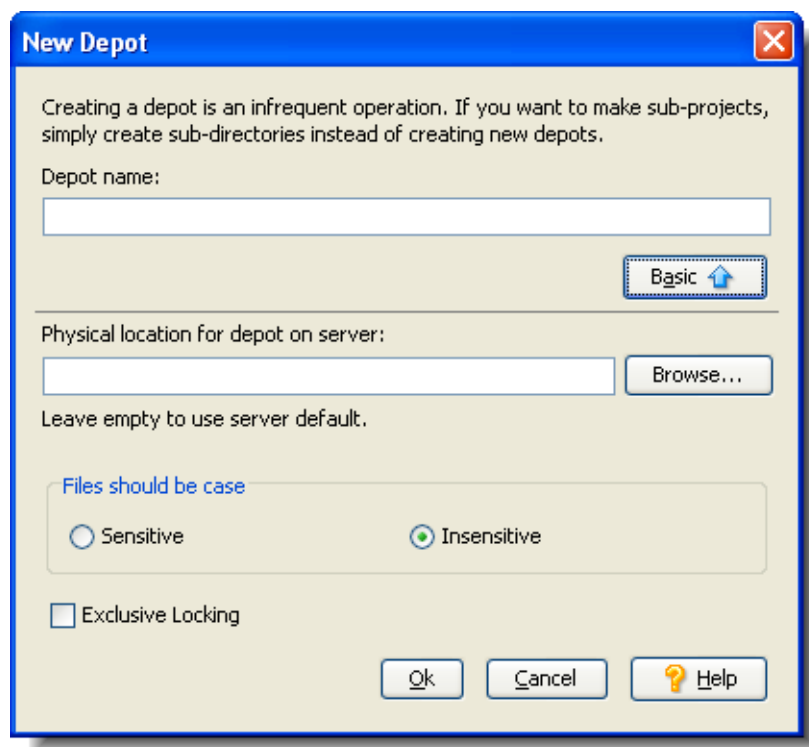
### Invoking the New Depot Command

You can invoke the *New Depot* command in any of these ways:

- Choose File > New > Depot from the GUI main menu.
- Choose Admin > Depots from the GUI main menu to open a Depots tab (See [“The Depots Tab”](#) on page 47.). Then click the New button.

### Using the New Depot Dialog

The *New Depot* dialog includes basic options (always visible) and advanced options (visibility controlled by a Basic/Advanced button).



### Basic Options

#### Depot Name

No two depots in the same AccuRev repository can have the same name. See [User-Specified Names for AccuRev Entities](#) on page 14.

*Note: Reusing a depot name*

This can be done—but it's risky (it cannot be undone) and it takes some work. You must first remove the existing depot with the desired name from the repository. For details, see "Removing a Depot from the AccuRev Repository" in the *AccuRev Administrator's Guide*.

## Advanced Options

### Physical location ...

The location you choose here must be on a hard drive on the machine where the *AccuRev Server* process runs. Be sure that the location does not overlap the location of any other depot, or any AccuRev *workspace* or *reference tree*. See [User-Specified Names for AccuRev Entities](#) on page 14.

If you do not specify a physical location, the new depot is placed under directory *storage/depots* within the AccuRev installation area.

A depot's physical location is said to be a *slice* of the overall AccuRev repository. To change the location of an existing depot, use the CLI command **chslice**.

### Files should be case ...

*Note:* We strongly recommend that you make your depots case-insensitive, for compatibility with Microsoft Windows. Some Windows applications can change the case of a filename without informing you; if your depot is case-sensitive, the file appears to be renamed.

All depots store names of files and directories exactly as they are originally entered (e.g. *WidgetGetCount* or *cmdListAll*). Likewise, AccuRev client programs display these names exactly as they were originally entered. The difference is that:

- A case-sensitive depot allows users to create two objects in the same directory, with names that differ only in their case (e.g. *makefile* and *Makefile*).
- A case-insensitive depot (default) does not allow two objects in the same directory to have names that differ only in their case.

A depot's case-sensitivity can be changed with the CLI **chdepot** command, as follows:

You can always change a depot's case from insensitive to sensitive.

You can only change from sensitive to insensitive if the depot contains no elements.

See [chdepot](#) in the *AccuRev CLI User's Guide* for more information.

### Exclusive locking

Selecting the *exclusive file locking* option causes all *workspaces* created for this depot to use AccuRev *exclusive file locking*.


See [File Locking](#) on page 259.


## The New Workspace Command


The *New Workspace* command creates a *workspace*, with a particular stream in a particular depot as its *backing stream*. The workspace can subsequently be *reparented* to another stream in the same depot.

### Invoking the New Workspace Command

You can invoke the *New Workspace* command in a number of ways:

- Choose File > New > Workspace from the GUI main menu.
- In an  Open Workspace dialog, click the New button.

Note: To open this dialog, select File > Open Workspace from the GUI main menu, or click the  button on the GUI main toolbar.

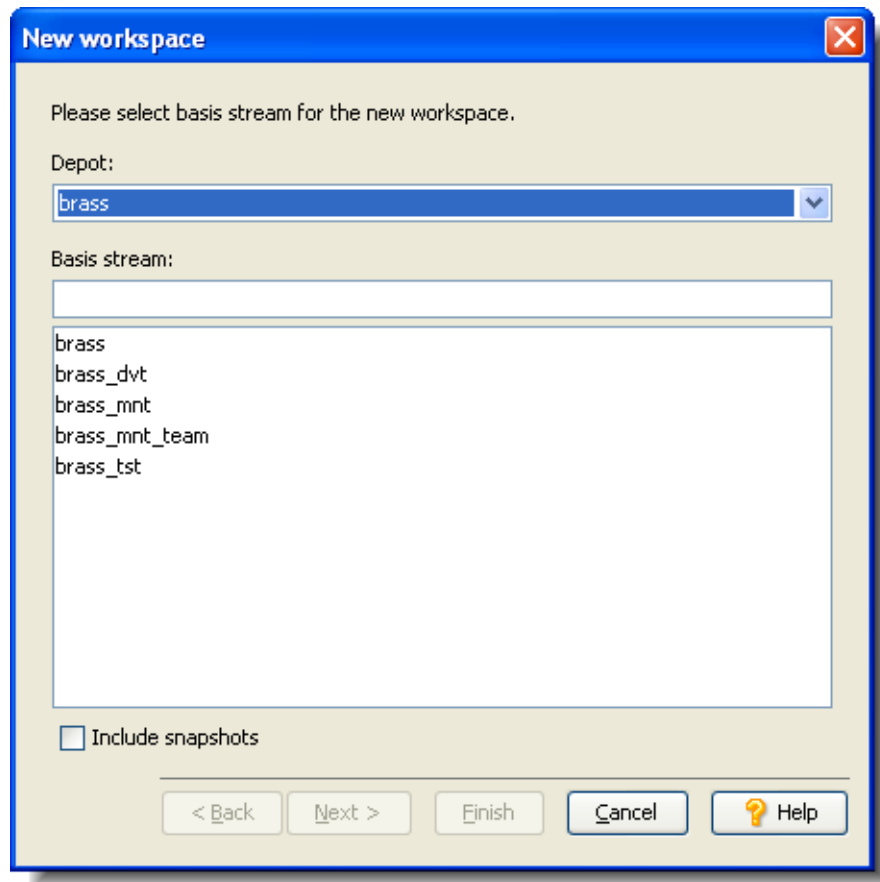
- Choose  New Workspace from the context menu of a stream or snapshot in the StreamBrowser.

### Using the New Workspace Wizard

AccuRev prompts you to specify the settings for the new workspace with a multiple-screen "wizard".

## The New Workspace Command: Screen 1 of 3— Specifying the Depot and Backing Stream

With the *File > New > Workspace* command, there may be no "current stream" context. In this case, AccuRev starts with this screen, prompting you to specify a stream context:



## Depot

Select one of the repository's depots. This populates the *Basis Stream* listbox with the streams (and optionally, snapshots) of the specified depot.

## Basis Stream

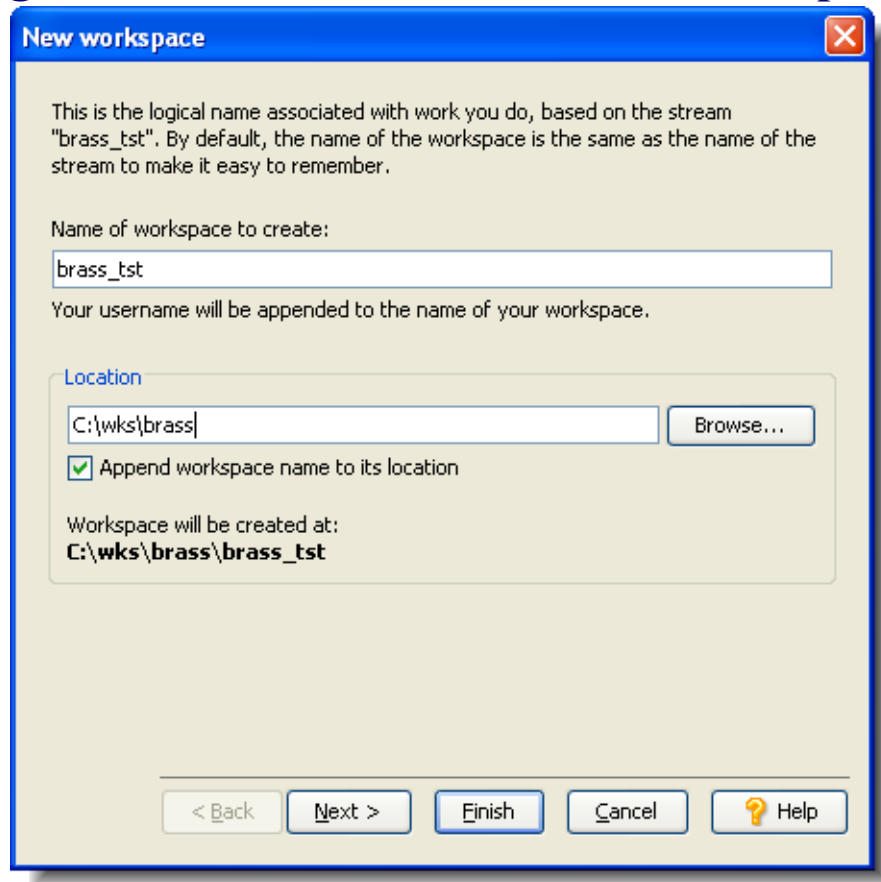
Select a stream or snapshot, to act as the backing stream for the new workspace.

*Note: Is it advisable to base a workspace on a snapshot?*

You can create new versions in a workspace based on a snapshot. But you won't be able to promote those versions to the snapshot (which is immutable, and so can't "accept" new versions). To get one or more of the versions out of the workspace, you can:

- **Reparent** the workspace to a dynamic stream, then **promote** the versions you've created in the workspace.
- Using the File Browser in another workspace, open one of the **version tools** on a particular element. Then use the Send to Workspace (See [“The Send to Workspace Command”](#) on page 132.) command to get the version that was created in the snapshot-based workspace.

## The New Workspace Command: Screen 2 of 3— Choosing a Name and Location for the Workspace)



### Name of Workspace to Create

No two workspaces in the same AccuRev repository can have the same name, even if they are in different depots. Moreover, streams, snapshots, workspaces, and reference trees all share the same namespace: no two of them can have the same name. See [User-Specified Names for AccuRev Entities](#) on page 14.

A workspace name always ends with a username suffix. For example, if your username is **akp** and you specify **thorn\_dvt** as the workspace name, AccuRev creates a workspace named **thorn\_dvt\_akp**.

#### Notes:

Can I type in the username suffix myself?

Yes. If your username is akp and you enter a workspace name that ends with \_akp, AccuRev doesn't change the name before creating the workspace.

Can I type in someone else's username suffix?

No, you cannot create a workspace for some other user in this way (or in any other way). If your username is akp and you enter the workspace name thorn\_dvt\_john, AccuRev creates a workspace named thorn\_dvt\_john\_akp.

### Location

You can specify any location in the client machine's file system for the new *workspace tree*. You can also create a workspace tree on a remote machine's disk, as long as that location is accessible on your machine. Access controls imposed by the operating system and/or network file system may restrict where you can create a workspace. See [User-Specified Names for AccuRev Entities](#) on page 14.

Here are some guidelines for using the *Append workspace name to its location* checkbox:

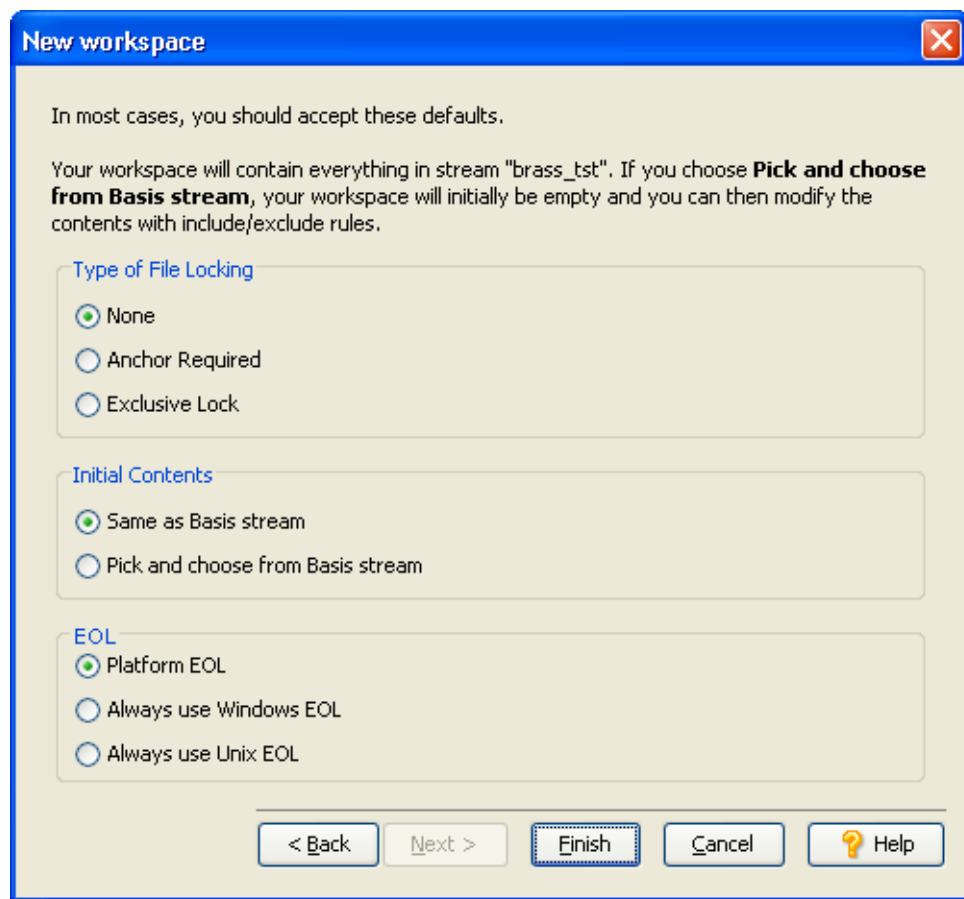
- If you are creating a new, empty workspace, enter the name of an existing directory (or Browse to it) and check the checkbox.
- If you are declaring that AccuRev should consider an existing directory to be a workspace, enter that directory's pathname and clear the checkbox.

In any case, make sure the *Workspace will be created at* value is correct before leaving this wizard screen.

*Notes:*

- Don't create a workspace within the AccuRev installation area — typically, [C:\Program Files\AccuRev](#) (Windows) or [/opt/accurev](#) (UNIX/Linux).
- Don't create a workspace within another workspace.
- Don't cross a physical file system boundary, whether local or remote. (For example: If you create a workspace at location [/opt/workspaces/derek](#), an error occurs if a remote file system is mounted at [/opt/workspaces/derek/src](#).)

## The New Workspace Command: Screen 3 of 3— Configuring the Workspace)



### Type of File Locking

This setting controls whether development in this workspace uses *serial* or *parallel* methodology. If you choose parallel development, you may still be restricted to using serial development on certain files; AccuRev supports locks on individual file elements. (See [File Locking](#) on page 259.)

**None (parallel development):** The version-controlled files in the workspace tree are writable at any time. There is no need to perform an *anchor* operation on a file before beginning to work on it.

**Anchor Required:** Each version-controlled file in the workspace tree is read-only until you use the *Anchor* command to make it writable. This also makes the file *active* in the workspace.

Note: The Anchor command makes the version currently in the workspace.

**Exclusive Lock:** Like *Anchor Required*, and also causes AccuRev to prevent users in *sibling* workspaces from working on the same file concurrently. (See [File Locking](#) on page 259.)

### Initial Contents

**Same as Basis stream** For each element, AccuRev copies the current version in the backing stream to the workspace tree. If no version of an element currently appears in the backing stream, it won't appear in the new workspace, either.

**Pick and choose from Basis stream:** AccuRev doesn't copy any files to the workspace tree. To begin working in the workspace, use the File Browser's Include/Exclude facility to configure the set of elements to appear in the workspace. ([Using the File Browser's Include/Exclude Mode](#) on page 102.)

### **EOL (Text-file line terminators)**

Whenever AccuRev copies a version of a text file from the repository to the workspace tree, it uses the line terminator you configure here.

**Platform EOL:** Use the line terminator for the operating system on the machine where the AccuRev GUI is running.

**Always use Windows EOL:** Use the sequence CR-LF (hex 0C-0A) as the line terminator.

**Always use UNIX EOL:** Use NL (hex 0A) as the line terminator.

## **The New Stream and Change Stream Commands (Stream Configuration dialog)**

The *New Stream* and *Change Stream* commands open a dialog in which you specify/revise the configuration of a new/existing stream.

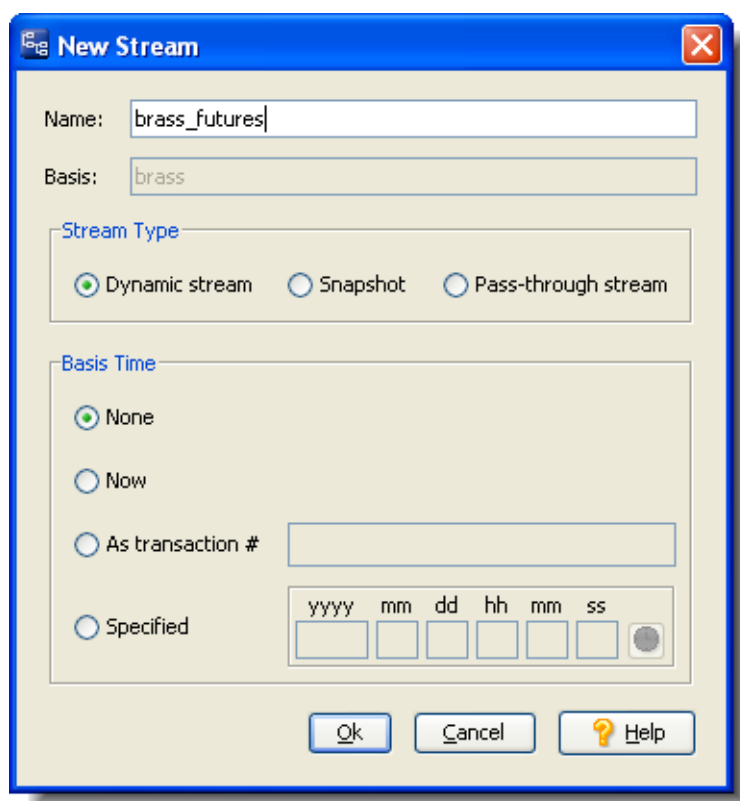
### **Invoking the Command**

In the StreamBrowser, right-click a stream to select it and display its context menu. Then,

- Choose New Stream from the context menu to create a new stream below the selected stream.
- Choose Change Stream from the context menu to revise the configuration of the selected stream.

### **Using the New Stream / Change Stream Dialog**

Enter or revise the following specifications, then click *Ok*.

The image shows a 'New Stream' dialog box with a blue title bar and a close button. It contains two text input fields: 'Name' with the value 'brass\_futures' and 'Basis' with the value 'brass'. Below these are two sections. The 'Stream Type' section has three radio buttons: 'Dynamic stream' (selected), 'Snapshot', and 'Pass-through stream'. The 'Basis Time' section has four radio buttons: 'None' (selected), 'Now', 'As transaction #' (with an empty text field), and 'Specified' (with a date/time picker showing 'yyyy mm dd hh mm ss' and a spin button). At the bottom are 'Ok', 'Cancel', and 'Help' buttons.

## Name

No two streams in the same AccuRev repository can have the same name, even if they are in different depots. See [User-Specified Names for AccuRev Entities](#) on page 14.

## Basis

The *parent stream* of the stream you are reconfiguring. Changing this entry *reparents* the stream when you click *Ok*.

In a *New Stream* dialog, this field is not editable, because you selected the parent stream when you invoked the command.

## Stream Type

The type of the stream:

- **Dynamic stream**
- **Snapshot**

Note: Selecting Snapshot is equivalent to invoking the command New Snapshot instead of New Stream.

- **Pass-through stream**

In a *Change Stream* dialog, this field is not editable. You cannot change the type of an existing stream.

## Date & Time

By default, a stream inherits the versions currently in its parent stream . If you assign a *basis time* to a stream, it inherits the versions that were in the parent stream at a specified point in time.

#### **None**

(disabled for *New Snapshot* or if you select Snapshot as the stream type) No basis time; in the future, the stream will inherit the versions currently in its parent stream .

#### **Now**


The stream or snapshot will contain the versions that were in its parent stream at the time you clicked *Ok* in this dialog.

#### **As of transaction #**

Enter an integer transaction number. The stream or snapshot will contain the versions that were in its parent stream at the time the specified transaction was created.

#### **Specified**

Enter a specific time. The stream or snapshot will contain the versions that were in its parent stream at the specified time.

You can fill in the individual subfields manually, or use the  *Select Date* button to display choices ("2 days ago", etc.) that fill in the subfields automatically. Once these subfields are filled in, you can revise them individually or by clicking the *Select Date* button again.

Note:

##### **1. Dynamic stream with basis time vs. Snapshot**

Setting a basis time on a stream does make it like a snapshot of its parent stream -- initially. But after that, new versions can be promoted to such a "time-based stream" from child workspaces and from other streams; you cannot promote new versions to a snapshot. Since you can change the basis time of a time-based stream as often as you wish, it can serve as a "moving snapshot" of its parent stream.

##### **2. Another way to control which versions are inherited from the parent stream**

The basis-time facility includes versions based on when they were created. The include/exclude facility includes elements based on their pathnames.

## **The Workspaces Tab / The Open Workspace Dialog**

The Workspaces tab displays summary information on some or all of the *workspaces* in the AccuRev repository.


The Open Workspace dialog, a slight variant of the Workspaces tab, makes it easy to open a File Browser tab on a selected workspace.

From either one, you can create a new workspace.

### **Opening a Workspaces Tab**

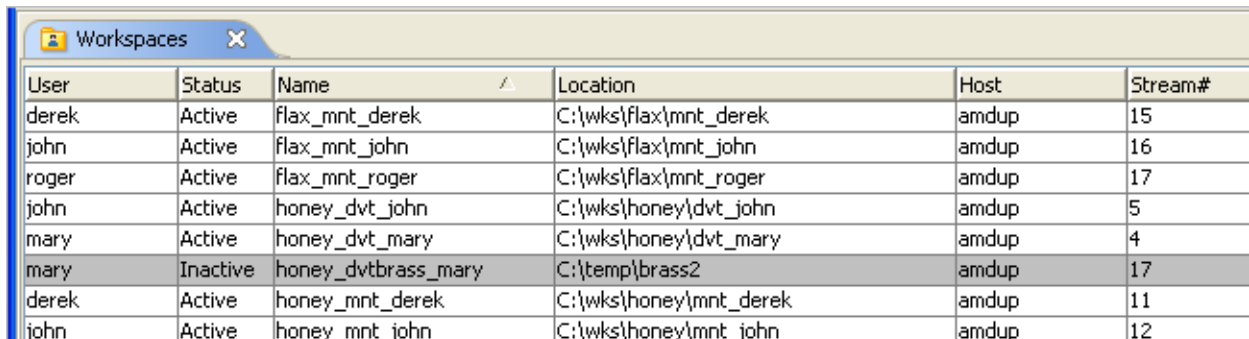
Choose *View > Workspaces* from the GUI main menu.

## Invoking the Open Workspaces Command

Choose *File > Open Workspace* from the GUI main menu, or click the  button in the GUI main toolbar.

## Workspaces Tab Layout

The Workspaces tab contains a table with these columns:



User	Status	Name	Location	Host	Stream#
derek	Active	flax_mnt_derek	C:\wks\flax\mnt_derek	amdup	15
john	Active	flax_mnt_john	C:\wks\flax\mnt_john	amdup	16
roger	Active	flax_mnt_roger	C:\wks\flax\mnt_roger	amdup	17
john	Active	honey_dvt_john	C:\wks\honey\dvt_john	amdup	5
mary	Active	honey_dvt_mary	C:\wks\honey\dvt_mary	amdup	4
mary	Inactive	honey_dvtbrass_mary	C:\temp\brass2	amdup	17
derek	Active	honey_mnt_derek	C:\wks\honey\mnt_derek	amdup	11
john	Active	honey_mnt_john	C:\wks\honey\mnt_john	amdup	12

### User

(appears if *Show all workspaces* is checked) The user who owns the workspace.

### Status

(appears if *Show including hidden* is checked) **Active** indicates that the workspace is available for use. **Inactive** indicates that the workspace is currently deactivated. Inactive workspaces can be reactivated. See the Remove and Reactivate commands below.

### Name

The simple name of the workspace, including the suffix that indicates the user who owns the workspace.

### Location

The pathname of the *workspace tree* on the machine where it resides.

### Host

The name or IP address of the machine where the workspace tree resides.

### Stream#

The integer stream-ID of the workspace. This number is unique with the workspace's depot only, not across the entire repository.

Note: More precisely, this integer identifies the private workspace stream that is an integral part of the workspace. A depot's streams, snapshots, and workspaces share the same "pool" of stream-IDs.

### Target Transaction

(appears if *Show details* is checked) The depot's *transaction level* at the time the *Update* command was most recently invoked on the workspace. *Update* attempts to load versions created in transactions up to and including the target transaction. If the **Target**

**Transaction** and **Update Level** are the same, the workspace's most recent update completed successfully.

### Update Level

(appears if *Show details* is checked) The highest-numbered transaction whose versions have been copied to the workspace in an *update*.

### Type

(appears if *Show details* is checked) An integer that indicates the type of workspace stream: 1 (standard workspace), 9 (exclusive-file-locking), or 17 (anchor-required).

### EOL

(appears if *Show details* is checked) The type of line terminator used when an update copies text-file versions to the workspace tree: **Auto** (line terminator used by the host machine's OS), **UNIX** (NL), or **Windows** (CR-LF).

## Working in a Workspaces Tab

You can apply filters to control the amount of data displayed in the workspaces table, and you can perform a number of operations on a selected workspace.

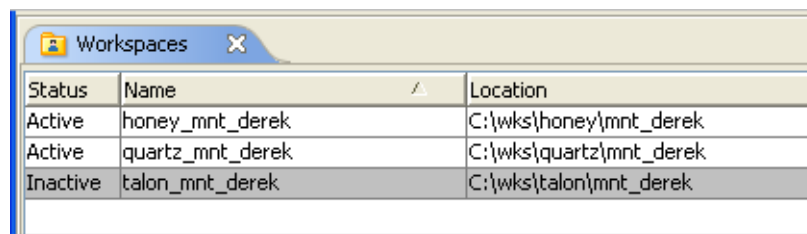
See also:

[Working with Tables](#) on page 9.

## Controlling Which Workspaces are Displayed

### Show including hidden

If checked, the table includes workspaces that have been deactivated with the Remove command. This also adds a Status column; deactivated workspaces have Inactive status, and are shaded.



Status	Name	Location
Active	honey_mnt_derek	C:\wks\honey\mnt_derek
Active	quartz_mnt_derek	C:\wks\quartz\mnt_derek
Inactive	talon_mnt_derek	C:\wks\talon\mnt_derek

### Show all workspaces

If checked, the table includes workspaces created by all users. This also adds a User column.

If cleared, the table includes your workspaces only.

## Controlling the Data Displayed for Each Workspace

### Show details

If checked, the table includes these columns: **Target Transaction**, **Update Level**, **Type**, and **EOL**.

## Operating on a Selected Workspace

You can choose any of the following commands from the context menu of a selected workspace:

### Open (Workspaces tab)

### Ok (Open Workspace dialog)

Using a File Browser tab, display the entire contents of the workspace.

### Edit

Modify the configuration of the workspace.

### Reactivate

Bring back into active service a workspace that was previously deactivated with the *Remove* command.

### Remove

Deactivate a workspace. The workspace cannot be used to perform AccuRev commands, although the workspace tree is not changed in any way. The workspace no longer appears in the StreamBrowser display (unless you use the StreamBrowser's *Show including hidden* option).

You can use the *Reactivate* command to bring back a *Remove'd* workspace.

## Creating a New Workspace

Click the *New* button at the bottom of the Workspaces tab to invoke the New Workspace wizard (see [The New Workspace Command](#) on page 18).

## The New Snapshot Command

The *New Snapshot* command opens a dialog in which you define of a *snapshot* of the stream you selected.

### Invoking the New Snapshot Command

In the StreamBrowser, right-click a stream and choose *New Snapshot* from the context menu.

Note: There is no "Edit Snapshot" command, because a snapshot cannot be modified in any way.

### Using the New Snapshot Dialog

The *New Snapshot* dialog is identical to the *New Stream* dialog (See [“Using the New Stream / Change Stream Dialog”](#) on page 23.), except that:

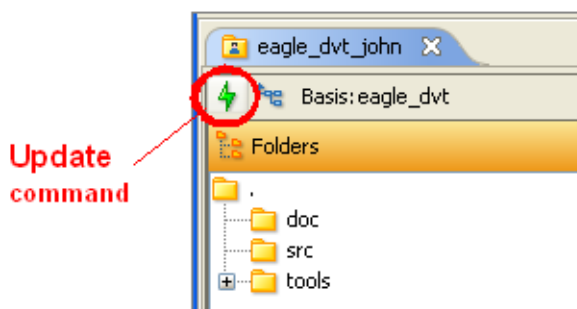
- The stream type Snapshot is preselected.
- You must specify a *basis time* (the None radio button is disabled).

## Updating a Workspace

The principal purpose of an AccuRev workspace is to provide an isolated location for performing development tasks. The changes you make do not affect your colleagues until you decide to make them public, using the *Promote* command. Likewise, the changes that others make do not

immediately affect your workspace. You must execute an *Update* command to bring versions created (and then promoted) by your colleagues into your workspace.

You invoke the *Update* command in the File Browser, using the *File > Update* command or a button above the Folders pane.



During execution of the *Update* command, a progress window appears (See [“The Update Progress Box”](#) on page 133.). A user preference controls how this window is used.) See [“AccuRev Preferences \(Tools > Preferences Command\)”](#) on page 42.

Prior to executing the *Update* command, you can use *File > Update Preview* or the Update Preview filter in the Searches pane to see what the Update will do. See [“File Browser: Working in the Searches Pane”](#) on page 59.

## Kinds of Changes Involved in an Update

At its simplest, an update just copies versions of some file elements from your workspace's *backing stream* into the workspace. For example, your colleagues may have edited the contents of files *colors.java* and *main\_menu.java*, created new versions of them in their workspaces, then promoted the versions to the common backing stream. When you invoke *Update*, the new versions of those two files are copied from the depot to your workspace, overwriting the older versions of the file.

In addition to incorporating such *content changes* into your workspace, *Update* incorporates *namespace changes* :

- renaming of a file
- moving of a file to another directory
- creation of new files and directories
- *defuncting* of existing files and directories

AccuRev tracks namespace changes in the same way as content changes -- by saving each change as a new version in your workspace. If you *Rename* file *colors.java* to *colours.java*, the change is recorded as a new version of the file. Changing the name again, to *hues.java*, creates another new version.

Similarly, defuncting file *rgb.java* creates a new version of the file in your workspace.

Note: See [“Renaming a Modified File Before Keeping It”](#) on page 126.

## How Update Works

The following sections provide an overview of how *Update* works.

## Deciding Which Elements to Update

The basic *Update* strategy is to leave the files you're working on undisturbed, and to copy any new versions of other files into the workspace. This enables the workspace to provide the safety of isolation, while still "keeping in touch" with other users' progress.

Roughly speaking, *Update* partitions the files in your *workspace tree* into two categories, to determine which are candidates for updating:

Files that you're currently working on. AccuRev uses the concept of **default group** to keep track of the files you're working on. Files are placed in the default group when you process them with such AccuRev commands as Send to Workspace, Keep, and Rename. These files are not candidates for updating, because it would overwrite your changes.

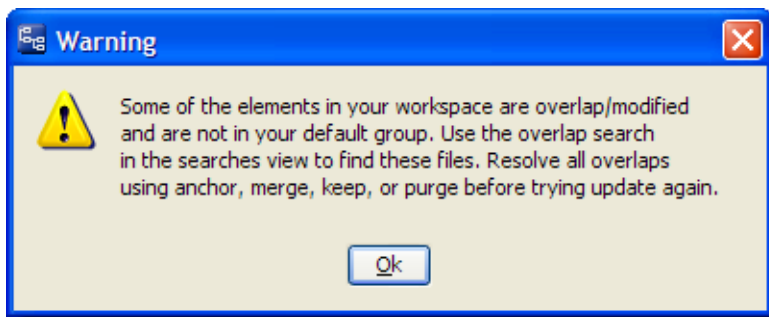
All the other files. The files that you're not currently working on are candidates for updating. AccuRev will update all such files for which a more recent version exists in the backing stream.

But things are not quite this simple. You might have edited other files, too, without preserving the changes with the *Keep* command. Such files have **(modified)** status.

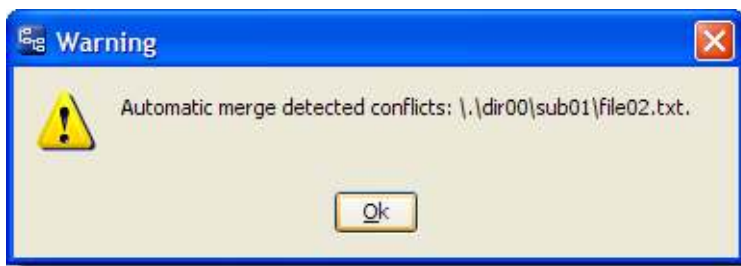
In many cases, no one else has been working on those files. If no new versions of the **(modified)**-status files have been created in the backing stream since your most recent *Update*, the update proceeds, leaving those files alone.

But suppose one or more of the files *does* have a new version in the backing stream. In this case, the file's status is **(modified)(overlap)**. *Update* won't simply overwrite the file with the backing-stream version, but it can attempt to *merge* the file with the backing-stream version:

- If user preference Update Resolves Trivial Merges is not selected, Update terminates immediately with an error box:



- If user preference Update Resolves Trivial Merges is selected, Update determines whether a merge can be performed automatically on all files with **(modified)(overlap)** status. If so, the update proceeds. If not, the update terminates with an error box:



*Notes:*

1. Members of the default group are "active" elements.

AccuRev documentation often uses the term **active** to describe the elements that are members of the default group.

2. Improving the performance of the non-member search.

To improve performance, **timestamp** and **pathname** optimizations enable Update to avoid examining every file in the workspace tree.

3. Anchor-required workspaces and exclusive file locking.

By default, AccuRev allows you to edit any file in a workspace at any time -- it doesn't require you to perform a "check out" operation on a file before editing it. This provides convenience and flexibility, but the edited files (with "non-member" status) can abort an Update. You can use the **anchor-required** workspace and/or **exclusive file locking** feature to ensure files never get "non-member" status, thus guaranteeing that Update will always proceed. See ["File Locking"](#) on page 259.

4. Determining which elements need to be updated: performance note.

This step is efficient and speedy. Update needs to consider only the elements that were involved in transactions recorded since the workspace's previous update. Only these transactions can contain changes that have not yet been incorporated into the workspace.

## Transferring Data from the Repository

*Update* next applies both content changes and namespace changes to elements in the workspace:

- It applies content changes to file elements by copying the backing-stream versions from the repository to the workspace tree. For (modified)(overlap)-status files, it merges the backing-stream version with the file in the workspace tree.
- It handles namespace changes to file and directory elements by creating, removing, renaming, and/or moving objects in the workspace tree.

## Recording the Update

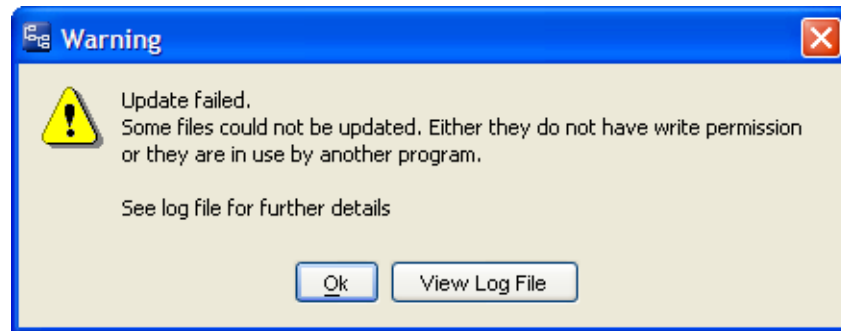
After it has completed the updating of versions in the workspace, *Update* changes the values of two workspace parameters:

- **scan threshold**: the more recent this value, the more effective the **timestamp optimization** used both by Update and by certain File Browser searches.
- **update level**: after an update, the workspace is "up to date as of transaction N "; N is the workspace's update level. The higher this value, the fewer transactions your next invocation of Update will need to examine, in order to determine which elements need to be updated.

## When Update does not Work

Sometimes AccuRev is not able to complete an *Update* operation, typically because a file cannot be overwritten due to permissions, or because it is open in another application. In this case, you

will see an "Update Failed" warning. Use the **View Log File** button on this dialog to display the details of the failure, which should allow you to troubleshoot and correct the issue.



## More on Update ...

For a more detailed description of how *Update* does its work, see section "The Update Algorithm" in *AccuRev Technical Notes*.

## The Properties Command

The *Properties* command displays a message box with information about the selected element.

### Invoking the Properties Command

Choose *Properties* from the context menu of an element in the Details tab of the File Browser.

### The Properties Display

#### File Path

The *depot-relative pathname* of the element in the current stream or workspace.

#### File Type

The *element type* of this version.

#### File EID

The integer that identifies this element. An element's simple name or pathname can change, but not its element-ID. No other element -- past, present, or future -- in the same depot can have this element-ID.

#### Version

The *version-ID* of this version.

#### File Status

The current *status* of this version.

#### Target Path

(*element link* or *symbolic link*) The *depot-relative pathname* of the link's target.

## The Issues Menu

See [The AccuWork Queries Tab](#) on page 287.

See [The Edit Form Tab](#) on page 273.

## The Tools Menu

### The Synchronize Time Command

The Synchronize Time command changes the system clock on the *client* machine to match the clock on the *AccuRev Server* machine. This succeeds only if the operating system grants you the right to make this change.

### The Show Info Dialog

The Show Info dialog displays overall information about your AccuRev work environment.

#### *Opening a Show Info Dialog*

Choose *Tools > Info* from the GUI main menu.

#### *Show Info Tab Layout*

#### **Principal**

Your AccuRev *user* identity, as established by AccuRev's *Login* command.

#### **Host**

The name or IP address of the client machine on which the AccuRev GUI is running.

#### **Server name**

The name or IP address of the server machine on which the AccuRev Server process is running. You can switch servers as part of the login process.

#### **Port**

The port number on which the AccuRev Server listens to client requests.

#### **ACCUREV\_BIN**

The directory on the client machine where AccuRev executables are stored. In older versions of AccuRev, you sometimes needed to create an environment variable with this name. This is no longer necessary.

#### **Client time**

The system time on the client machine. (The parenthesized integer is the number of seconds since Jan 1, 1970.)

#### **Server time**

The system time on the server machine. (The parenthesized integer is the number of seconds since Jan 1, 1970.)

If the client time and server time differ by more than 5 seconds, many AccuRev commands fail to execute. Use the *Tools > Synchronize Time* command to fix this problem.

# The Server Tasks Tab

The Server Tasks tab lists the current set of "worker threads" in the multi-threaded *AccuRev Server* process. At any given moment, each worker thread is executing one of the server **subtasks** that implement a particular client command. Each command from a *client program* translates to at least two server subtasks.

## Opening a Server Tasks Tab

Choose *Tools > Server Tasks* from the GUI main menu.

## Server Tasks Tab Layout

The Server Tasks tab contains a table with the following columns:

### Task ID

The ID number of the client-server connection. Each client command establishes a separate connection with the server.

### Task Name

The name of the server subtask being executed by this worker thread. This typically matches the name of the client command, such as Update.

### User

The AccuRev username of the user who issued the client command.

### Host

The IP address of the client machine where the user invoked the client command.

### Status

0 = slot assigned to worker thread, but thread not active yet (set by master thread).

1 = worker thread active (set by worker thread).

2 = worker thread completed, but master thread has not yet cleared the slot for reuse (set by worker thread).

### Start Time

The time at which the worker thread started working on this subtask.

### Seconds on Server

The number of seconds ("wall clock" time) that this worker thread has been running.

*Note: Command-line interface to this command*

You can use the AccuRev CLI to list the Server Tasks tab data:

1. Create a text file, for example tasks.xml, with this one-line XML document:

```
<tasklist/>
```

2. Execute this AccuRev CLI command:

```
accurev xml -l tasks.xml
```

## Working in the Server Tasks Tab

The Server Tasks tab display is informational only. See also: See [“Working with Tables”](#) on page 9.

## The Login Command

The *Login* command establishes your AccuRev *user* identity. In the "User Name" field, you must enter the name of a registered AccuRev user . If your organization has more than one AccuRev *repository* (each managed by a separate *AccuRev Server* process), you can choose a particular Server to connect to.

Note: Active and inactive users. A user name can be Inactive (Remove User command) or Active (when first created; Reactivate User command). When you log in, the username must be currently Active.

See also: [Security](#) on page 255

## Invoking the Login Command

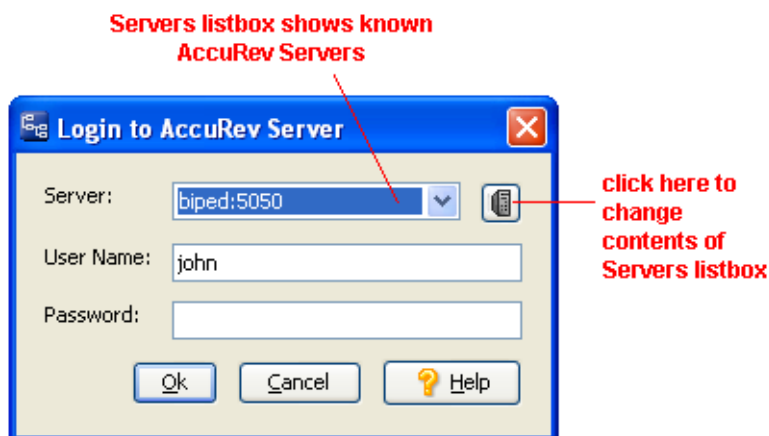
Choose *Tools > Login* from the GUI main menu. This command is invoked automatically ...

- at the beginning of an AccuRev GUI session if there is no valid *session file* in the .accurev subdirectory of your home directory
- when you invoke a command that requires user authentication, but your session has expired

## Using the Login Dialog

### To login

Type in your username -- and if you have one, your password -- then click **Ok**. Passwords are case-sensitive.





### To switch to another server

Open the listbox by clicking the down-arrow, then select another AccuRev Server (hostname/port-number combination).

Then, complete the login by typing your username (and password).

### To add an AccuRev Server to the listbox

Click the  button to display the *Available AccuRev Servers* dialog. (See [“Using the Available AccuRev Servers Dialog”](#) on page 269.). Click the  *Add Server* button to specify:

- The hostname (or IP address) where the additional AccuRev Server runs.
- The IP port number on which it listens for AccuRev client requests.

## The "session" File

A successful *Login* command creates an encrypted file in subdirectory *.accurev* of your home directory. This file records your AccuRev username and password, along with the IP address of your client machine. Most AccuRev client commands can be executed only by an authorized user. Such commands send the information in your session file to the AccuRev Server process, so that you don't need to repeatedly "remind" the AccuRev Server who (and where) you are.

## Session Expiration

Session files are automatically deleted by the AccuRev Server after an administrator-configurable interval, which defaults to 240 minutes. This implements a session-expiration feature.

## Multiple Session Files for Different Servers

The name of the session file includes the hostname and port number of the AccuRev Server. If you *Login* to different AccuRev Server processes, you'll have multiple session files, one for each Server. Example:

```
session_VENUS_5050
session_MARS_5050
session_MARS_5999
```

These session files indicate that you are logged in to an AccuRev Server on host *venus*, listening on port 5050, and are also logged into two different AccuRev Server processes on host *mars*.

## Multiple Session Files for the Same Server

What if you want to be testuser *john* in one GUI window, but testuser *mary* in another -- both using the same AccuRev Server? You can't have two session files with the same name (for example, *session\_VENUS\_5050*) in your *.accurev* subdirectory. But you can have two or more session files with the same name in different *.accurev* subdirectories. The AccuRev GUI uses the value of environment variable *ACCUREV\_HOME* as the location of the *.accurev* subdirectory (and will create this subdirectory, if necessary). It also uses *ACCUREV\_HOME* whenever it checking for the existence of a session file in order to authenticate you.

Example (Windows):

1. Set *ACCUREV\_HOME* to *c:\myusers\john*.
2. Create directory *c:\myusers\john*.
3. Start the AccuRev GUI.
4. Login as *john*.
5. Set *ACCUREV\_HOME* to *c:\myusers\mary*.
6. Create directory *c:\myusers\mary*.
7. Start another instance of the AccuRev GUI.
8. Login as *mary* in this second instance.

Hint: Setting up scripts to run the AccuRev GUI makes it easier to ensure that the GUI will be running with the environment variable set correctly.

## The Change Password Command

The *Change Password* command changes (or creates) the password for your AccuRev *user* identity.

See also: [Security](#) on page 255.

### Invoking the Change Password Command

Choose *Tools > Change Password* from the GUI main menu.

### Using the Change Password Dialog

Type the new password twice, once in the Password field, and again in the Confirm Password field.

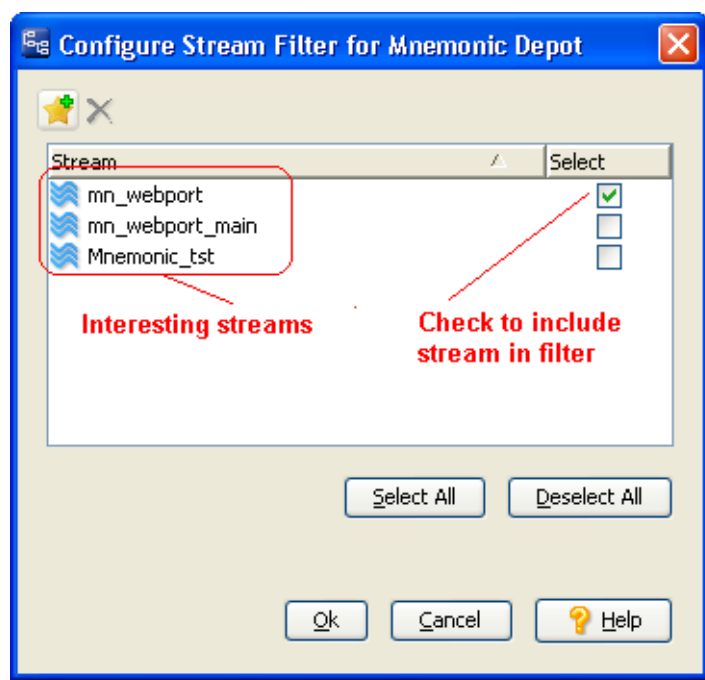


The change takes place only if your two entries match exactly.


## Filter Streams (Tools > Filter > Streams Command)

### Invoking the Filter Streams Command

You can invoke this command in any of these ways:



• Choose *Tools > Filter > Streams* from the GUI main menu.

- Click the  Filter Streams button on the GUI main toolbar.
- You also have the option to bypass the dialog and add or remove streams directly from the StreamBrowser's context menu. See the [Quick Stream Filtering Options](#) on page 40 for more information.

## About the Stream Filter

If your organization has depots containing thousands of streams, the *stream filter* is a powerful tool to improve system performance and focus attention on just the streams you're interested in.

Adding a stream to the filter will limit the *stream hierarchy* shown in the StreamBrowser to the *stream path* of that stream and all its children. Also, any list of streams available in the GUI will contain **only** the subset of streams shown in the StreamBrowser.

## Using the Configure Stream Filter Dialog

The Filter Streams command allows you to restrict the total number of streams displayed in the AccuRev GUI. When you invoke the command, the Configure Stream Filter dialog shows a list of all the streams that you have chosen. You can use this dialog to keep a list of "interesting" streams, and to control which streams are part of the *stream filter*. A stream is part of the stream filter if its checkbox is selected in the list shown in this dialog.

You can use the dialog to:

- Add more streams to the list
- Remove streams from the list

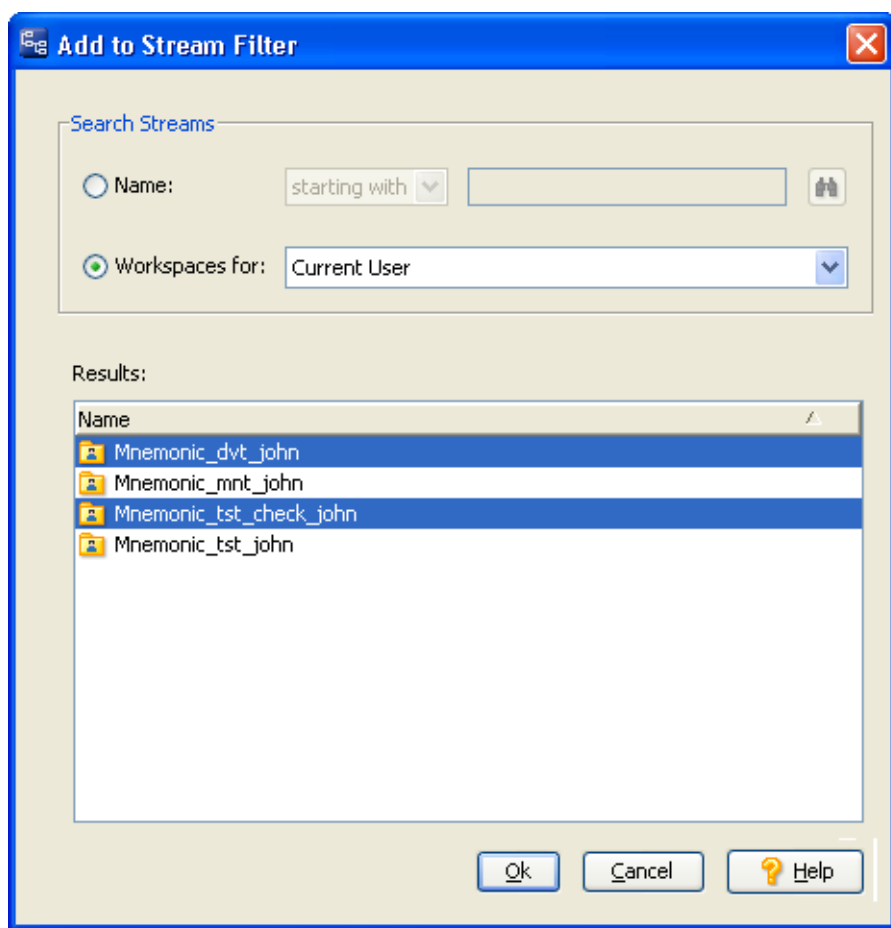
- Toggle the checkbox(es) next to one or more streams to indicate that they should be added to or removed from the stream filter. If you have many streams of interest, you may find it useful to use the Select All and Deselect All buttons to quickly make your selections.

After making your selections, click *Ok* to filter streams in the AccuRev GUI.

Note: Stream filters are applied first to the data displayed in the Stream Browser. Any use of the display controls in the StreamBrowser is applied to the filtered set of streams. Occasionally, the interaction between these two sets of controls may produce a "surprising" result. For example, if the stream filter contains only one user's workspaces, and the Workspaces drop-down list is set to show another user's workspaces, the (correct) result is a StreamBrowser showing only the root stream. If your stream filter does not produce the expected display in the Stream Browser, remember to check the display controls to verify that these local settings are not affecting the outcome.

## Adding Streams

You can add streams to the stream filter by clicking the  *Add to List* button.




The resulting dialog allows you to search in two ways for streams that you would like to add:

- by stream name
- by workspace

When the Results list contains the desired streams, select one or more and click *Ok* to add them to the *stream filter*.

Note: If you add the root stream to the stream filter, all streams in the depot will be shown, since all streams are children of the root stream.

## Searching by Stream Name


Use the *Name* radio button to search by stream name. Stream name searches can be exact matches (using *matching*), or partial matches (using *starting with*, *ending with*, or *containing*). Once you have specified the item to search for, click the  *Search* button or press *Enter* to see the resulting list of streams in the Results list.

## Searching by Workspace

Use the *Workspaces for* radio button to search by workspace. Choose a user or group name from the drop-down list to see the list of those workspaces in the Results list.

Note that the users and groups available from the drop-down list are limited by any user/group filters you have set.

## Removing Streams

You can remove streams from the list in the Configure Stream Filter dialog by selecting one or more streams and clicking the  *Remove from List* button. Click the *Ok* button to save your changes.

This action only removes the streams from the list in the dialog; it does not apply the *Remove* command to the stream itself. If the stream you remove is part of the *stream filter*, the StreamBrowser display (and lists of streams in the GUI) will reflect that change.

## Quick Stream Filtering Options

In the *StreamBrowser*, there are two methods of quickly filtering the stream display:

- Add a stream directly to the stream filter. To do this, right-click on the stream, and choose the Add to Stream Filter option from the context menu that appears. This action will immediately apply the filter, causing the StreamBrowser display to change, so if you have multiple streams to add to the filter, you may want to use the Filter Streams command instead.


Note: As an alternative, you can add the root stream to the filter first, so that all streams will display. Then add streams to the filter one-by-one, and remove the root stream from the filter when you are finished.

- For a single user or group, show only the stream paths to that user's or group's workspaces. To do this, select a user or group from the Workspaces drop-down list at the bottom of the StreamBrowser, then select the Only stream paths to workspaces checkbox. This option does not add the workspaces to the stream filter, nor does it restrict the display of streams anywhere but the StreamBrowser; it is simply a shortcut display mechanism there.

## Filter Users/Groups (Tools > Filter > Users/Groups Command)

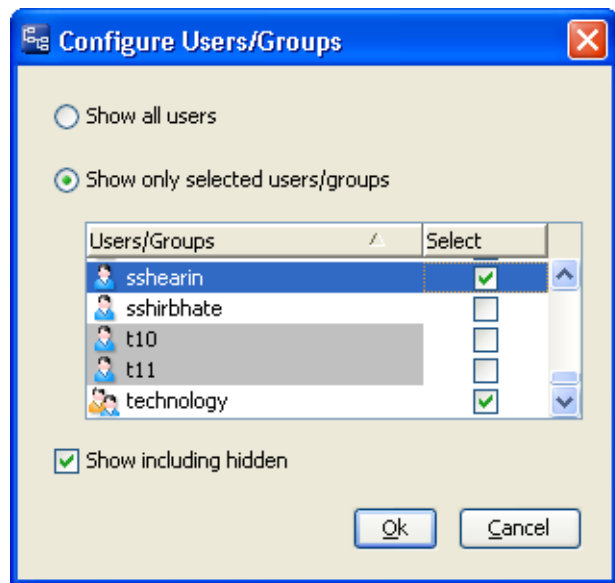
### Invoking the Filter Users/Groups Command

You can invoke this command in any of these ways:

- Choose Tools > Filter > Users/Groups from the GUI main menu.
- Click the Filter Users/Groups button  on the main GUI toolbar.

### Using the Dialog

The *Tools > Filter > Users/Groups* command allows you to restrict the set of users and groups displayed in certain places in the AccuRev GUI.



When you invoke the command, a dialog shows a list of all the users and groups you have defined within AccuRev. You can:

- Click the Show all users radio button to remove any restrictions on the display of users.
- Click the Show only selected groups/users radio button to enable filtering. Then click the checkbox in the Select column for each user or group you would like to display. When you have finished your selections, click the Ok button to enable the filter.

Use the *Show including hidden* checkbox to include users that have been deactivated with the *Remove* command in the list shown in this dialog. These users are displayed with a grey background.

### Scope of the Users/Groups Filter

The users and groups you specify will appear in lists of users in the following locations:

- Workspaces drop-down list in the StreamBrowser.

- All drop-down lists in AccuWork that contain user or group data (such as Assigned to or Submitted by), and throughout the workflow.
- Add to Stream Filter dialog.
- Send to Issue dialog (when change-package-level integration is set up between AccuRev and AccuWork).


## AccuRev Preferences (Tools > Preferences Command)

The *Tools > Preferences* command opens a tabbed dialog, in which you can manage a set of parameters that influence the way the AccuRev looks and works. The preferences you set here apply only to you. They are stored in file *preferences.xml*, in subdirectory *.accurev* of your home directory. (By default, AccuRev determines your home directory by making a call to the operating system. If environment variable ACCUREV\_HOME is set, AccuRev uses the value of this variable as your home directory.)

### General Page

#### Enable Issue Preview

Controls the automatic inclusion of an Edit Form pane below the Query Results pane in a AccuWork Queries tab. The contents of the issue record that is currently selected in the Query Results pane is automatically displayed in the Edit Form pane.

The toolbar of the Query Results pane includes a  *Show Issue Form* button, which toggles the inclusion of the Edit Form pane.

Despite the name "Preview", the Edit Form pane is fully functional: you can revise, save, and export the contents of issue records in this pane.

**Note:** This option will be disabled if your AccuRev Java client is configured to display issues in the AccuRev Web UI. This will be the case if:

- AccuWorkflow is enabled.
- Your AccuRev Administrator has configured your site to use the Web UI for AccuWork.
- You have set your local preference to "Open Issues In Web Only". (See next section.)

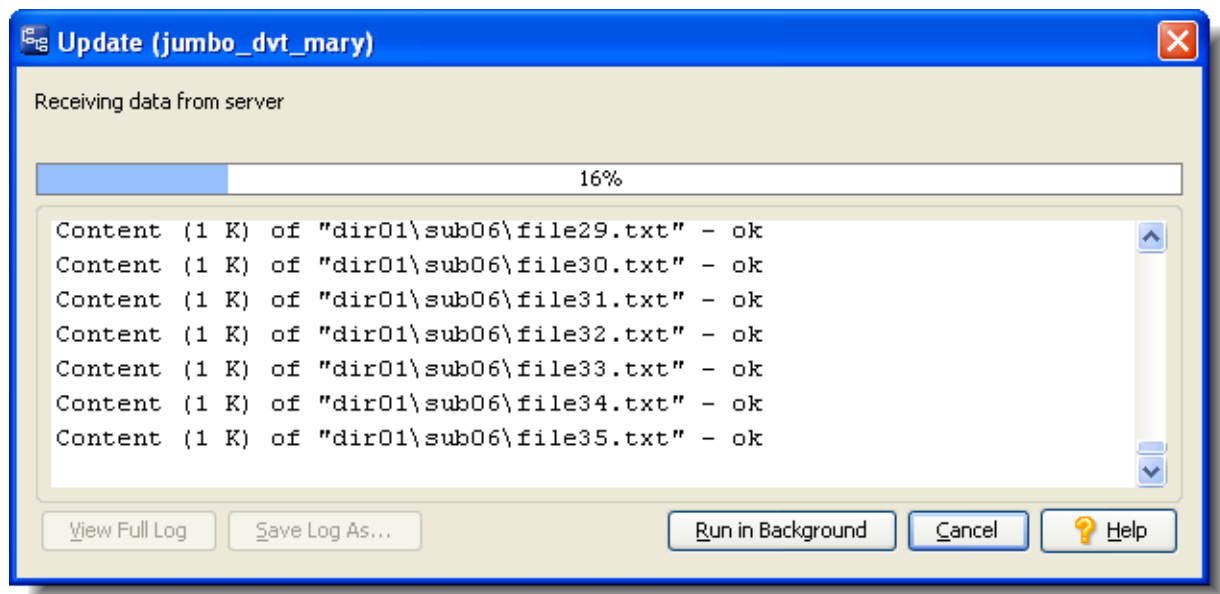
#### Open Issues in Web Only

If this option is available, and if you set it, the AccuRev Web UI will be used whenever you access an AccuWork issue. This option may already be set for you if AccuWorkflow is enabled at your site, or if your AccuRev Administrator has set this option site-wide.

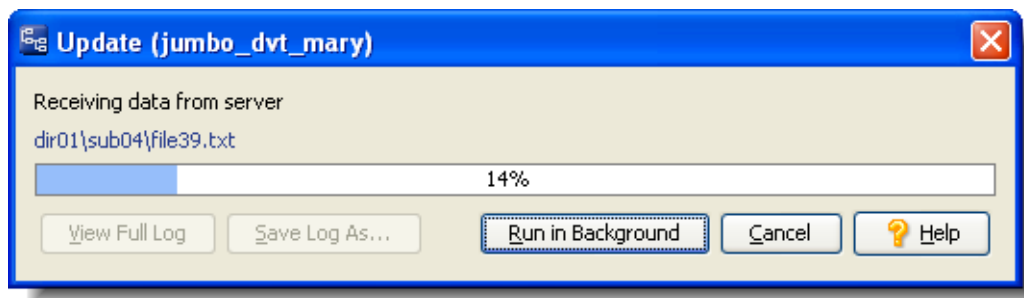
#### Show Progress Log

Controls whether the *Update* or *Update Preview* dialog includes a scrollable text field that displays the names of up to 2000 elements that are being updated (or would be updated).

If this option is checked, the dialog looks like this:



If this option is cleared, the dialog looks like this:



After the command finishes its work, a complete log is always available, through the *View Full Log* button in the dialog.


### Show External Elements

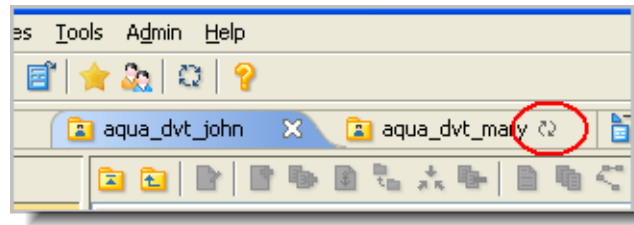
Controls whether the File Browser's Details pane includes files and directories that have not been placed under version control (*Add to Depot* command). This setting applies only when you're working in the Folders pane.

### Confirm On Exit

Controls whether the AccuRev GUI prompts for confirmation when you invoke the *File > Exit* command or attempt to shut the GUI window.

### Require Manual Refresh

If this preference is cleared, the data displayed on a "buried" GUI tab is automatically refreshed, if necessary, when you switch to that tab. The AccuRev GUI signals that a refresh is necessary with a  "refresh" icon next to the tab title.



If this preference is checked, the data on a "buried" tab is not refreshed automatically when you switch to it. Use the *View > Refresh* command (or function key **F5**) to refresh the data.

### Keep Session Active

Controls whether the session file in the *.accurev* subdirectory of your home directory will be deleted when you end the AccuRev GUI session. This file is created by the *Tools > Login* command, and is used by other AccuRev commands to establish your identity (and the identity of your client machine).

### Use Ignore Element Optimization

Causes the *Update* command to use the same pathname optimization that is used by certain File Browser searches.

### Update Resolves Trivial Merges

Enables the *Update* command to perform a merge operation on text file elements that are not in a workspace's default group, but have **(overlap)(modified)** status. The merges are performed and the *Update* proceeds only if for *all* such files, the merge is completely automatic (there are no *conflicting changes*).

### Display of element name in tables

Controls how element pathnames are displayed in tables: in a single *Element* column, or in separate *Name* and *In Folder* columns.

### Online Help Browser

The full pathname of the web browser to be used for displaying the GUI's context-sensitive help screens. This field also determines what web browser to use if your client is configured to work with the AccuRev Web User Interface (Web UI).

## Diff/Merge Page

The **Diff/Merge** tab allows you to specify supported third-party tools to use with AccuRev. **Note:** Any tools that you specify must be defined in your PATH variable before launching the Java GUI. If you specify a tool and it does not work, this is the most common cause.

### Diff

Specifies the graphical file-comparison tool to be invoked by the GUI's *Diff* command. You can choose AccuRev's own tool or one of the supported third-party tools from the

combo-box. Alternatively, you can type a command line to be executed when the *Diff* command is invoked. This command line must include substitution patterns:

**%1%** Quoted pathname of the first version to be compared.

**%2%** Quoted pathname of the second version to be compared.

**%3%** Unquoted title string for the first version (can be displayed by file-comparison tool).

**%4%** Unquoted title string for the second version (can be displayed by file-comparison tool).

Make sure the file-comparison tool is located in a directory on your search path. Alternatively, use a full pathname to specify the file-comparison tool. Examples:

**MyDiff /G %1% %2%**

**C:\SuperCompare\SCgui %1% %2%**

Notes:

What's the difference between Ignore Whitespace and Ignore Changes in Whitespace?

Ignore Whitespace causes all SPACES, TABs, and empty lines to be ignored, and thus overrides any setting of Ignore Changes in Whitespace.

Ignore Changes in Whitespace applies only where whitespace already exists; it does not apply where new whitespace is introduced or existing whitespace has been completely deleted.

Example 1: When comparing these two text strings:

```
compareme
compare me
```

Table 1.

<input type="checkbox"/> Ignore Whitespace	<input type="checkbox"/> Ignore Changes in Whitespace	Does Diff see difference?
X	either	No
-	X	Yes
-	-	Yes

Example 2: When comparing these two text strings:


```
compare      me
compare me
```

Table 2.

<input type="checkbox"/> Ignore Whitespace	<input type="checkbox"/> Ignore Changes in Whitespace	Does Diff see difference?
X	either	No
-	X	No
-	-	Yes

I changed my preferences, but I don't see any changes in the Diff tab.

Changing preferences does not automatically affect Diff tabs that are currently open. To apply the preferences, refresh the tab in any of these ways:

- Click the  Refresh button on the GUI window's main toolbar.
- Run the View > Refresh command on the GUI window's main menu.
- Press function key F5.

## Merge

Specifies the graphical text-file merge tool by be invoked by the GUI's *Merge* command. This command line must include substitution patterns:

**%a%** quoted filename of the closest common ancestor version.

**%1%** quoted filename of the version in the backing stream, or other non-workspace version.

**%2%** quoted filename of the version in the workspace.

**%3%** unquoted title string for the closest common ancestor version (can be displayed by text-file-merge tool).

**%4%** unquoted title string for the backing-stream version (can be displayed by text-file-merge tool).

**%5%** unquoted title string for the workspace version (can be displayed by text-file-merge tool).

**%o%** quoted name of the merge-output file -- a temporary file for storing the results of the merge.

Make sure the text-file-merge tool is located in a directory on your search path. Alternatively, use a full pathname to specify the text-file-merge tool.

*Note: Do I have to use a 3-way merge tool?*

No. You're free to use a 2-way merge tool, which takes into account the two contributor versions only, not any of their ancestors.

But even if you don't use a 3-way merge tool, AccuRev records the merge operation (which appears as a red line in the **Version Browser**). Future merges involving this element can use this record to simplify the work needed to perform a 3-way merge.

## Tab size

(AccuRev's graphical Diff and Merge tools only) The number of spaces to be displayed for each *TAB* character.

## Character Encoding

(AccuRev's graphical Diff and Merge tools only) Specifies the character encoding of the contributor versions: either UTF-8 or the operating system default.

## Ignore Whitespace

(AccuRev's graphical Diff tool only) Controls whether whitespace is taken into account when comparing text lines.

### Ignore Changes in Whitespace

(AccuRev's graphical Diff tool only) Controls whether a change in the amount of whitespace in a text line is considered to be a change to that line.

### Ignore Case

(AccuRev's graphical Diff and Merge tools only) Controls whether uppercase and lowercase characters are considered to be the same when comparing text lines.

## Version Browser Page

### Initial display mode




Specifies the mode that a new Version Browser tab begins in. *Basic* mode displays "important" versions only, for example omitting intermediate versions in workspace streams. *Expanded* mode displays all versions. Buttons at the bottom of the Version Browser tab switch between the two display modes.

### Initial transaction count

How many of the most recent transactions (and equivalently, versions) a new Version Browser should display. Controls in the Version Browser toolbar change this count.

## StreamBrowser Page

### Display Default Group in Stream Browser

Controls whether the StreamBrowser includes a current development activity control ( or  or ) for each stream and workspace with *active* elements. Disabling this feature can significantly improve StreamBrowser performance.

### Enable StreamBrowser History

Controls the inclusion in the StreamBrowser toolbar of controls that enable you to "turn back the clock", viewing a depot's stream hierarchy as it existed at any point in the past. You must refresh any existing StreamBrowser tabs to make a new setting effective.

## The Admin Menu

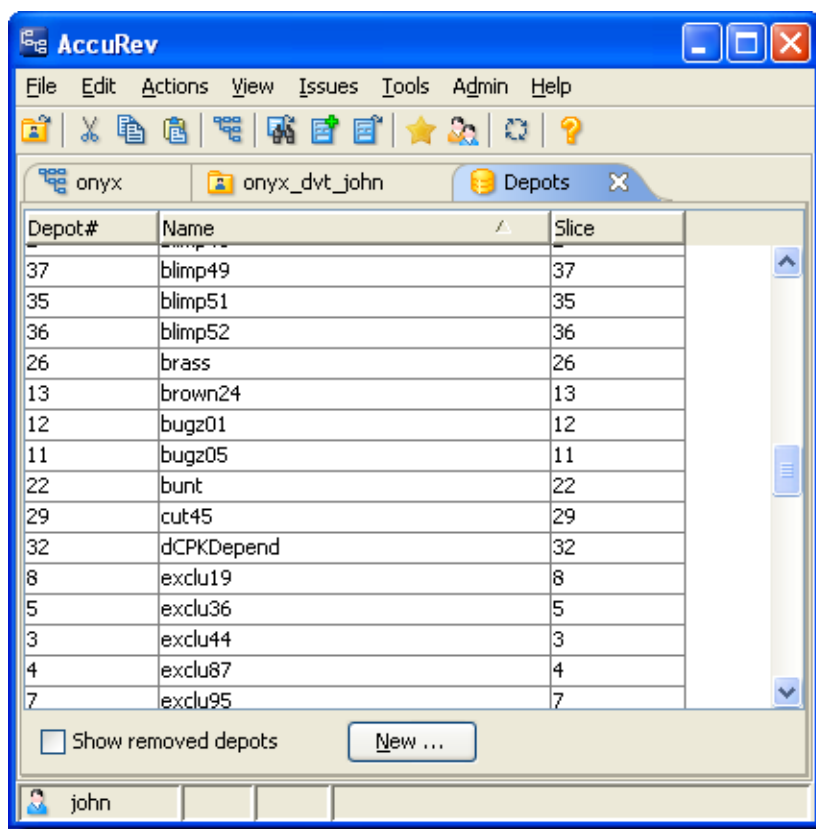
### The Depots Tab

The Depots tab lists some or all of the *depots* in the AccuRev *repository*.

### Opening a Depots Tab

Choose *Admin > Depots* from the GUI main menu.

## Depots Tab Layout



### Status

(appears if *Show removed depots* is checked) **Active** indicates that the depot is available for use. **Removed** indicates that the depot has been permanently removed from the repository. (The **maintain rmdepot** command permanently and irreversibly removes a depot from the repository. See "Removing a Depot from the AccuRev Repository" in the *AccuRev Administrator's Guide*.)

### Depot#

The unique integer that identifies a particular depot. This identifier cannot be modified.

### Name

The depot's name, which must differ from all other depot names in the repository. A depot can be renamed, but its depot number never changes.

### Slice

Same as Depot#.

## Working in a Depots Tab

You can control the display of removed depots.

See also: [Working with Tables](#) on page 9

## Controlling Which Depots are Displayed

### Show removed depots

If checked, the table includes the depots that have been permanently deactivated with the **maintain** utility (see the AccuRev *Administrator's Guide*). This also adds the *Status* column to the table; removed depots have **Inactive** status, and are shaded.

### Creating a New Depot

#### New

Click this button to invoke the [The New Depot Command](#) on page 16 .

## The Reference Trees Tab

The Reference Trees tab displays summary information on some or all of the *reference trees* in the AccuRev repository.

### Opening a Reference Trees Tab

Choose *Admin > Reference Trees* from the GUI main menu.

### Reference Trees Tab Layout

The Reference Trees tab contains a table with these columns:

#### Status

(appears if *Show including hidden* is checked) **Active** indicates that the reference tree is available for use. **Inactive** indicates that the reference tree is currently deactivated (but can be reactivated).

#### Name

The name of the reference tree.

#### Location

The pathname of the reference tree on the machine where it resides.

#### Host

The name or IP address of the machine where the reference tree resides.

#### Stream#

The integer stream-ID of the stream on which the reference tree is based. This number is unique with the reference tree's depot only, not across the entire repository.

#### Target Transaction

(appears if *Show details* is checked) The depot's *transaction level* at the time the *Update* command was most recently invoked on the reference tree. *Update* attempts to load versions created in transactions up to and including the target transaction. If the **Target Transaction** and **Update Level** are the same, the reference tree's most recent update completed successfully.

#### Update Level

(appears if *Show details* is checked) The highest-numbered transaction whose versions have been copied to the reference tree in an *update*.

### Type

(appears if *Show details* is checked) The integer 3, indicating that the type of data structure is a reference tree.

### EOL

(appears if *Show details* is checked) The type of line terminator used when an update copies text-file versions to the reference tree: **Auto** (line terminator used by the host machine's OS), **UNIX (NL)**, or **Windows (CR-LF)**.

## Working in a Reference Trees Tab

You can apply filters to control the amount of data displayed in the reference trees table, and you can perform a number of operations on a selected reference tree. See also: [Working with Tables](#) on page 9

### Controlling Which Reference Trees are Displayed

#### Show including hidden

If checked, the table includes reference trees that have been deactivated with the *Remove* command. This also adds a *Status* column; deactivated reference trees have **Inactive** status, and are shaded.

### Controlling the Data Displayed for Each Reference Tree

#### Show details

If checked, the table includes these columns: **Target Transaction**, **Update Level**, **Type**, and **EOL**.

### Operating on a Selected Reference Tree

You can choose any of the following commands from the context menu of a selected reference tree:

#### Open

In this release, the *Open* command is disabled. You cannot view the contents of a reference tree using the AccuRev GUI.

#### Edit

Modify the configuration of the reference tree.

#### Reactivate

Bring back into active service a reference tree that was previously deactivated with the *Remove* command.

#### Remove

Deactivate a reference tree. The reference tree cannot be *Update*'d.

You can use the *Reactivate* command to bring back a *Remove*'d reference tree.

## Creating a New Reference Tree

Click the *New* button at the bottom of the Reference Trees tab to invoke the [Using the New Workspace Wizard](#) on page 51.

## The New Reference Tree Command

The *New Reference Tree* command creates a *reference tree* for a specified *stream*.

### Invoking the New Reference Tree Command

1. Choose *Admin > Reference Trees* from the GUI main menu, to open a Refs tab.
2. Click the *New* button at the bottom of the tab.

### Using the New Workspace Wizard

The wizard for creating a new reference tree is essentially similar to the [The New Workspace Command](#) wizard.

## The Triggers Tab

The Triggers tab displays the *triggers* defined for a specified *depot*.

See also: [Security](#) on page 255.

### Opening a Triggers Tab

Choose *Admin > Triggers* from the GUI main menu.

### Triggers Tab Layout

The Triggers table includes these columns:

#### Type

**pre-create-trig, pre-keep-trig, pre-promote-trig, or server-post-promote-trig.**

#### Execute

The simple name or pathname of the script (or other program) that executes when the trigger fires.

### Working in a Triggers Tab

The *Triggers* tab is informational only. All trigger maintenance is performed with the AccuRev CLI. See the description of the *mktrig* and *rmtrig* commands. The data in the Triggers tab is the same as that displayed by the CLI command *show triggers*.

## The Slices Tab

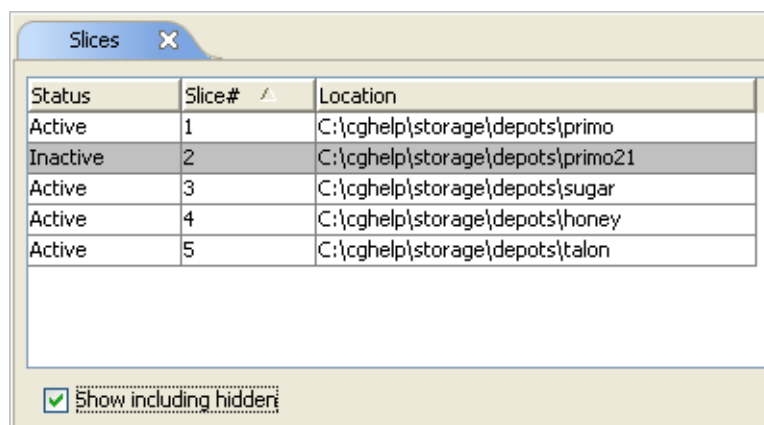
The Slices tab shows the physical storage locations of the depots in the AccuRev repository. If the repository is replicated, it shows the locations of the depots in the local replica repository only.

## Opening a Slices Tab

Choose *Admin > Slices* from the GUI main menu.

## Slices Tab Layout

The Slices tab contains a multi-column table.



Status	Slice#	Location
Active	1	C:\cghelp\storage\depots\primo
Inactive	2	C:\cghelp\storage\depots\primo21
Active	3	C:\cghelp\storage\depots\sugar
Active	4	C:\cghelp\storage\depots\honey
Active	5	C:\cghelp\storage\depots\talon

☒ Show including hidden

### Status

(appears if *Show including hidden* is checked) **Active** indicates that the depot is available for use. **Inactive** indicates that the depot has been permanently removed from the repository. (The **maintain rmdepot** command permanently and irreversibly removes a depot from the repository. See "Removing a Depot from the AccuRev Repository" in the *AccuRev Administrator's Guide*.)

### Slice#

The unique integer that identifies a particular depot. This identifier cannot be modified.

### Location

The pathname of the slice's top-level directory. This location can be changed -- for example, to move the depot to a larger and/or faster disk.

## Working in a Slices Tab

You can control the display of removed depots. See also: [Working with Tables](#) on page 9.

### Controlling Which Slices are Displayed

#### Show including hidden

If checked, the table includes the slices of depots that have been permanently deactivated with the **maintain** utility (see the *AccuRev Administrator's Guide*). This also adds a *Status* column; removed depots have **Inactive** status, and are shaded.

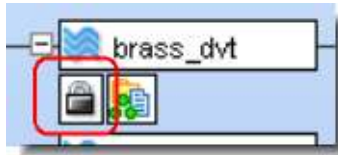
## The Locks Command

The *Locks* command maintains the set of locks that apply to a *dynamic stream*. Locks control the ability to Promote versions to/from the stream, the ability to configure the stream's contents with the *include/exclude facility*, and the ability to modify the stream's settings.

See also: [Security](#) on page 255.

### Invoking the Locks Command

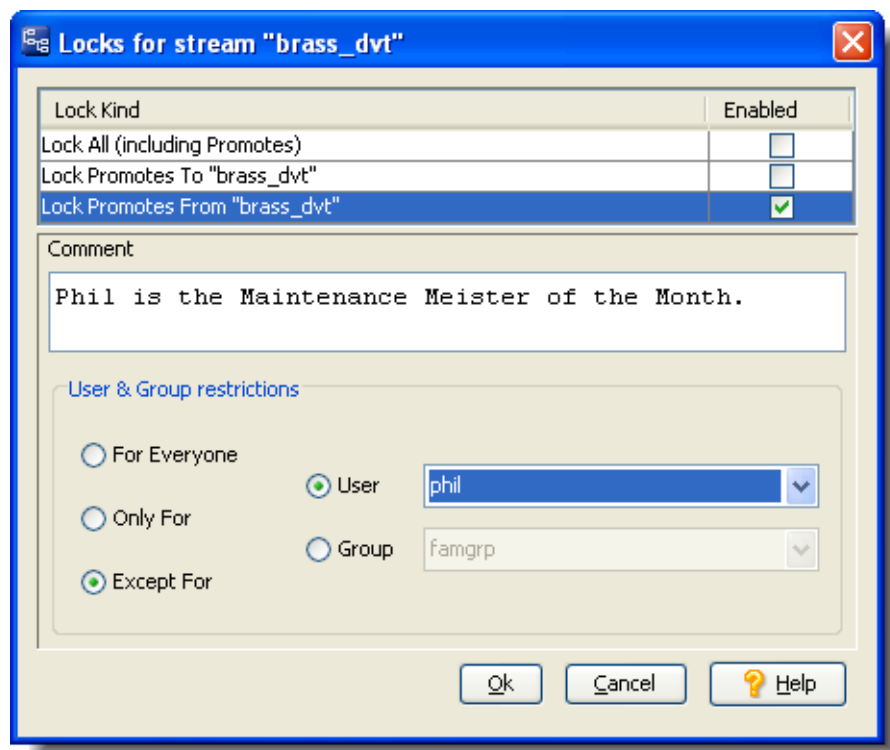
You can open a Locks dialog in the *StreamBrowser* in any of these ways:



- Right-click a dynamic stream, and select *Locks* from the context menu.
- (if there are currently one or more locks on the stream) Click the *Locks* icon that appears below the stream

### The Locks Dialog Box

On a given stream, you can create one lock of each type:



#### Lock All

Disables promotion of versions to/from the stream; disables include/exclude mode changes to the stream; disables the Change Stream command for the stream. (Does not disable the Remove command.)

**Lock Promotes To**

Prevents versions from being added to the specified stream's default group. Disables promotion to the stream.

**Lock Promotes From**

Prevents the removal of versions from the stream's default group. Disables promotion and purging from the stream.

A *Lock All* setting takes precedence over "to" and/or "from" settings.

You can restrict the effect of the lock:

**Only for**

The lock applies only to the specified user, or to all the members of the specified group.

**Except for**

The lock applies to all users, except for the specified user, or except for all the members of the specified group.

## 3. The File Browser

AccuRev's job is to keep track of your files. Accordingly, one of AccuRev's main GUI tools is the **File Browser**. The File Browser makes it easy to view, monitor the status, and change the contents of *elements* located in AccuRev *workspaces*. You can also use it to view and monitor the status (but not modify) of elements in *streams* and *snapshots*.

You can have any number of File Browser tabs open concurrently in an AccuRev GUI window -- each one displaying the contents of a difference workspace, stream, or snapshot.


See Also:

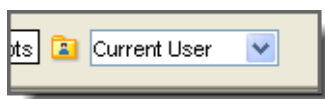
[Alternatives to the File Browser](#) on page 57

[File Browser: Browsable Data Structures](#) on page 58

### Opening a File Browser Tab

There are several ways to open a File Browser tab in the AccuRev GUI. When you start a new GUI session, one or more File Browser tabs (and other tabs) might open automatically, enabling you to continue from where you ended the previous session. To open a File Browser on any of your workspaces, use one of these techniques:


- Select *File > Open Workspace* from the command menu (or click the  *Open Workspace* toolbar button). Select a workspace and click *Ok*, or double-click it.
- On a StreamBrowser tab, make sure the desired workspace is displayed. To open another user's workspace, change the listbox setting at the bottom on the tab from *Current User* to the appropriate *username*.



- There is also an *All Workspaces* setting. Right-click the desired workspace, and select *Open* from the context menu. Or just double-click the desired workspace.

A new File Browser tab opens with the top-level folder in the Folders pane selected.

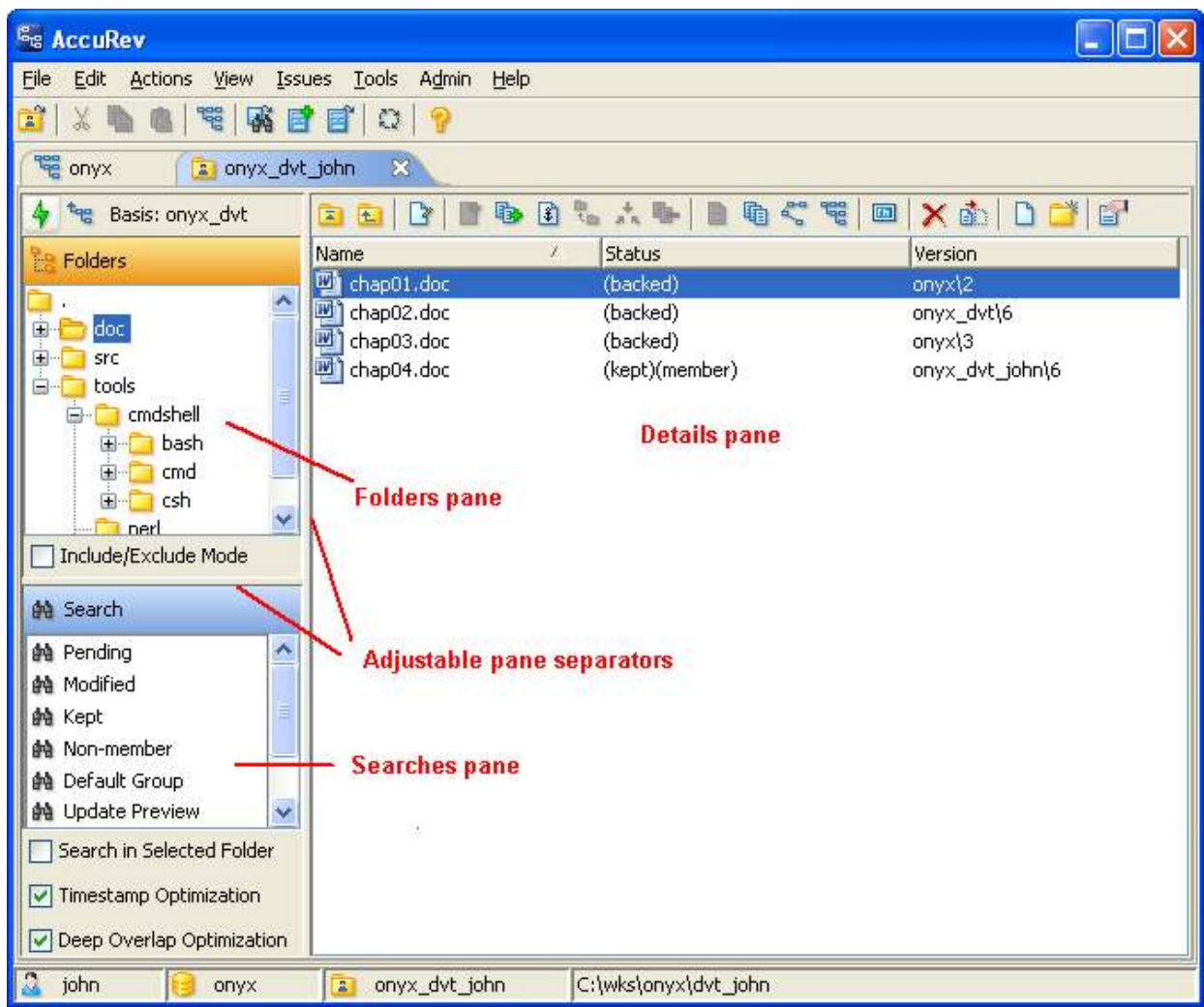
### Leaving and Returning to a File Browser Tab

You can leave a File Browser, switching to another tab in the AccuRev GUI, then return to that File Browser tab later. When returning to a File Browser tab, it's a good idea to refresh the display (with **View > Refresh**, the  toolbar button, or function key **F5**). This ensures that the File Browser display reflects any work you performed "between visits" to the tab.

Alternatively, you can close the tab altogether: right-click the title, then select *Close* from the context menu. You can also close the tab using the "X" icon on the tab control itself.

### File Browser Tab Layout

The File Browser display resembles that of Windows Explorer. To the familiar Folders pane and Details pane, it adds a unique Searches pane. You can execute AccuRev commands by selecting from the GUI window's main menu, by clicking toolbar buttons, or by using the context (right-click) menus of items in the File Browser display.



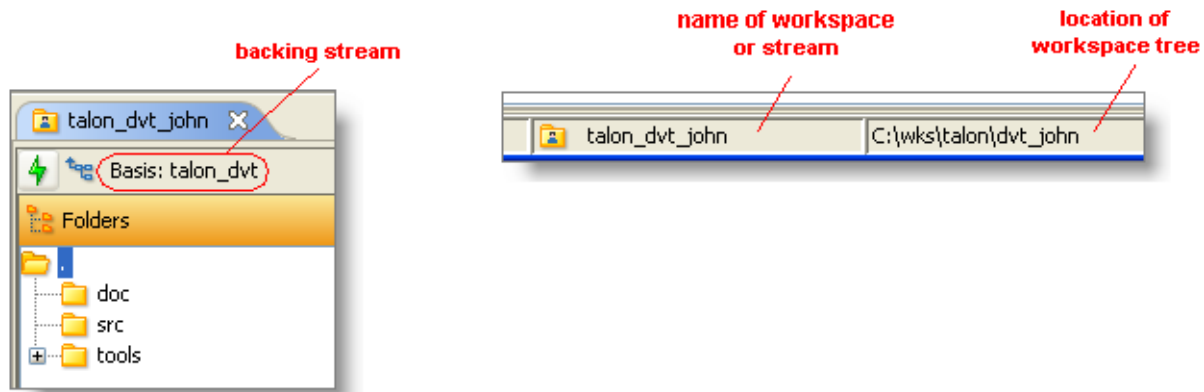
The *Folders pane* enables you to navigate the folder (directory) hierarchy of a workspace or stream. When you're using the Folders pane, the *Details pane* displays the contents of the currently-selected folder.

The *Searches pane* lists element-status searches that you can apply to the entire workspace or stream. When you're using the Searches pane, the Details pane displays all the elements that meet the selected search criterion. (In the AccuRev CLI, the search facility is described using the term "filter".) For example, it can display all the files you've edited and saved with the *Keep* command.

## GUI Window Indicators

When you're working in the File Browser, the indicators at the bottom of the AccuRev GUI window show the name of the current workspace or stream. For a workspace, it also shows the

location of the *workspace tree* in disk storage. The name of the workspace's *backing stream* is shown at the top of the Folders pane.



## Working in the File Browser's Panes

Click on the links below for more information on working in the File Browser's panes.

- [File Browser: Working in the Folders Pane](#) on page 58
- [File Browser: Working in the Searches Pane](#) on page 59
- [File Browser: Working in the Details Pane](#) on page 67
- [Updating a Workspace](#) on page 28

## Alternatives to the File Browser

The File Browser is not intended to completely replace the Windows Explorer or comparable tools on UNIX systems. For example, the File Browser does not attempt to implement the Windows "file types" functionality, based on filename suffixes.

Some people do most of their work inside an integrated development environment (IDE), such as Visual Studio or Eclipse. These environments have their own "explorer" or "file browser" built in, typically organizing files into separate "projects". You can use AccuRev commands within an IDE if an AccuBridge integration for that environment is available (or an integration from a third-party developer). Even if you use an IDE most of the time, you may want to use the File Browser occasionally:

- The IDE does not display all of a file's configuration-management properties, such as the current version-ID.
- Certain AccuRev commands cannot be executed through the IDE integration, only from the AccuRev File Browser.

It's important to keep in mind that you can use a wide variety of tools to make *content changes* to a file, but you must use AccuRev commands to make valid *namespace changes*: renaming, moving to another directory, or deleting. The AccuBridge IDE integrations support these namespace operations.

## File Browser: Browsable Data Structures

You can open a File Browser tab on these AccuRev data structures, each of which provides access to a *configuration* of your organization's *version-controlled* files and directories:

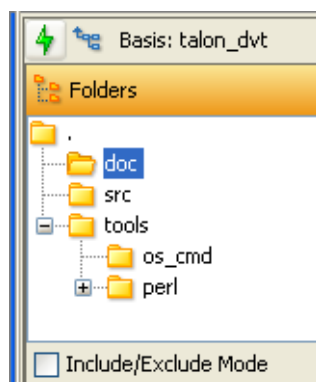
- *workspace*
- *stream*
- *snapshot*

AccuRev also supports a data structure similar to a workspace, called a *reference tree*. You cannot open a File Browser tab on a reference tree. (But the *Admin > Reference Trees* command displays the references trees created for a particular depot.)

For a full discussion of these data structures, see the *AccuRev Concepts Manual*.

## File Browser: Working in the Folders Pane

The Folders pane includes a tree control, for navigating the folder (directory) hierarchy of a workspace.

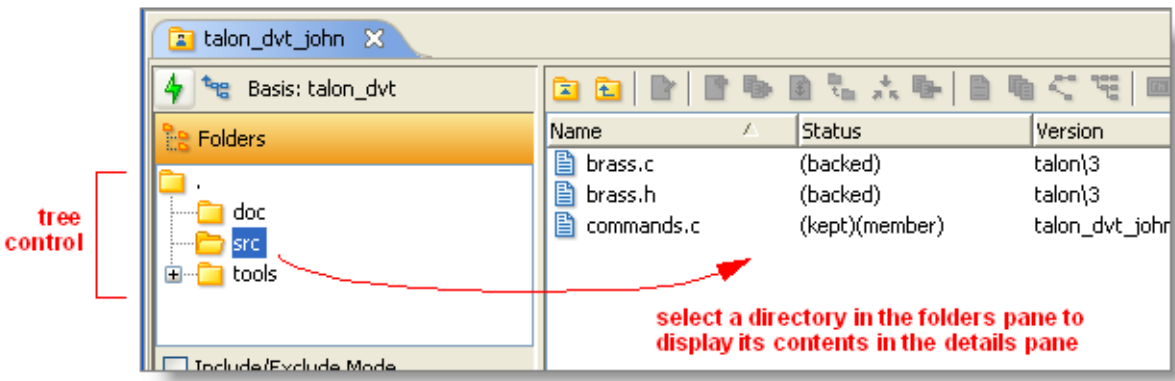


It works in the standard way:

- A "+" (or right-arrowhead) control indicates that the folder has a subhierarchy that is not currently displayed. Left-click this control to open the subhierarchy.
- A "-" (or down-arrowhead) control indicates that the folder has a subhierarchy that is currently displayed (in part, or in its entirety). Left-click this control to close the subhierarchy.

You can also double-click the folder icon or the folder name next to a control to open or close it.

When you select a particular folder (directory), the objects in that folder appear in the Details pane.



## Working in Include/Exclude Mode

The set of directories that appears in the Folders pane depends on whether the File Browser is currently in include/exclude mode (see [Using the File Browser's Include/Exclude Mode](#) on page 102):

- When you're in include/exclude mode, the Folders pane displays the depot's entire directory hierarchy (as it is inherited from higher-level streams). This enables you to change your include/exclude configuration.
- When you leave include/exclude mode, the File Browser applies the include/exclude rules; the Folders pane displays only the directories that are configured to be included in the workspace or stream.

## Folders Pane Command Reference

You can invoke the following commands from the Folders pane toolbar.



### Update

Update the workspace or reference tree.



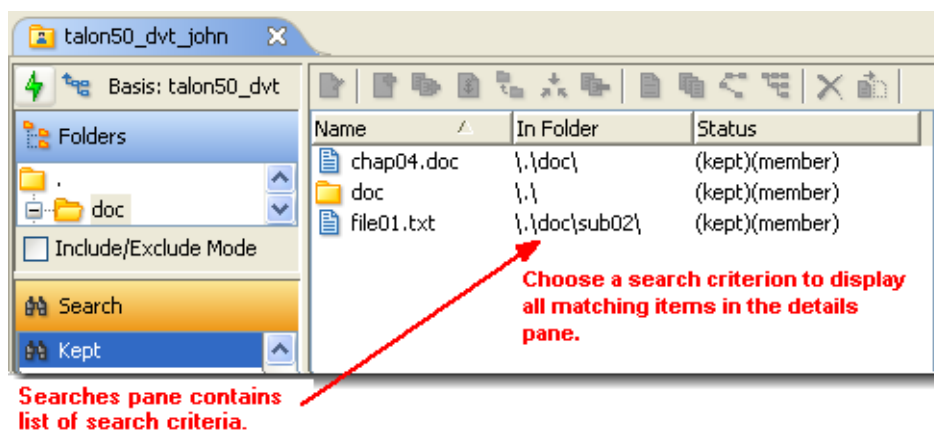
### Open Basis Stream

Open the basis (backing) stream in a separate File Browser tab.

## File Browser: Working in the Searches Pane

The File Browser can organize the contents of a workspace or stream by AccuRev *status*, instead of by directory location. (You can also search by element name. See [File Browser: Searching by Element Name](#) on page 65 for details.) When you click one of the items in the Searches pane (e.g. *Kept*), the File Browser searches the entire workspace or stream for objects that meet that search

criterion, and displays those elements in the Details pane. You can perform commands on these objects, using the toolbars and menus.



File Browser searches enable you to see "just the files you care about", instead of all the files in your workspace. (The AccuRev CLI uses the term "filter" instead of "search".) Which files do you care about? Roughly speaking, a workspace contains a copy of its backing stream -- often, the entire source base -- which might include hundreds or thousands of files. But for a given development project, you'll probably modify only a handful of the files. You may not need the other files at all; or you may need them for general reference, or to enable you to perform software builds and tests in the workspace.

The searches are not mutually exclusive: some objects may be selected by more than one search criterion -- for example, *Kept* and *Pending*: every object selected by a *Kept* search will also appear in the results of a *Pending* search.

See also:

[AccuRev Element Status](#) on page 89


[Using the File Browser's Include/Exclude Mode](#) on page 102

## Performing Searches of the Entire Workspace or Stream

To perform a search of the entire workspace, dynamic stream, or snapshot on which you opened the File Browser, just click one of the search criteria in the Searches pane. AccuRev performs the search and displays the resulting set of elements in the Details pane.

You can perform as many searches as you like; the File Browser remembers the results of the most recent search of each kind. So, for example, you might:

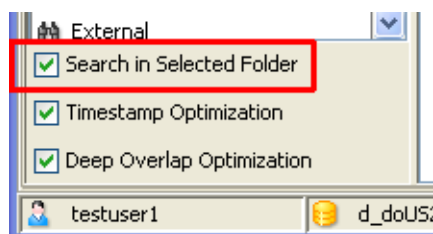
1. Click the Kept search criterion to search for files with **(kept)** status in your workspace.
2. Click the Overlap search criterion to search for files with **(overlap)** status in your workspace.
3. Click the Kept search criterion again.

Instead of performing another search, the File Browser just displays the results of your previous search for files with **(kept)** status. You can refresh the display (with **View > Refresh**, the  toolbar button, or function key **F5**) to have the File Browser perform the Kept search again, in order to take into account recent changes.

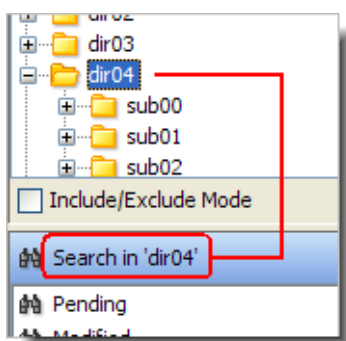
The cache of search results is maintained until you close the File Browser tab.

## Restricting a Search to a Subtree

The *Search in Selected Folder* checkbox at the bottom of the Searches pane enables you to restrict any search to the subtree of the folder that is currently selected in the Navigation pane.

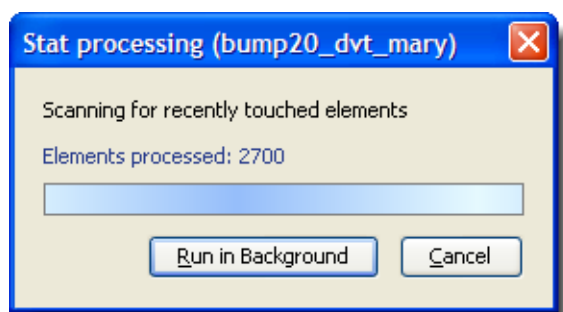


If you check or clear this checkbox when the Details pane already contains the results of a search, AccuRev automatically performs the search again in the restricted (or unrestricted) context. If Search in Selected Folder is checked, the folder context appears in the Searches pane banner:

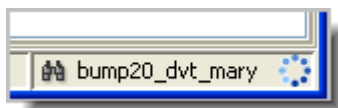


## Search Progress Box

A search can be almost instantaneous or not, depending on the number of elements in the workspace or stream. During a longer search, AccuRev displays a progress box.



If you click the Run in Background button, the progress box is replaced by an indicator in the lower right corner of the GUI window:



The indicator disappears when the search completes.

## Search Criteria

The Searches pane contains these search criteria:

### **Pending**

(workspace only) Select elements whose status includes either **(modified)** or **(kept)**. These are elements that have had a *content change* or a *namespace change*.

### **Modified**

(workspace only) Select files whose status includes **(modified)**. This indicates a content change that has not yet been preserved with *Keep* (but may have followed a previous *Keep*).

### **Kept**

(workspace only) Select files whose status includes **(kept)**. These are files whose content changes have all been preserved with *Keep*, or elements with namespace changes (*Rename* or *Defunct* command).

### **Non-member**

(workspace only) Select files whose status includes **(modified)** but not **(member)**. These are files that have content changes, but are not in the workspace's default group. They have not been activated with a *Keep* or *Anchor* command since the file's last update or promotion.

### **Default Group**

Select elements whose status includes **(member)**. These are elements in the workspace's default group, for which you've entered one of these commands: *Keep*, *Rename*, *Defunct*, *Anchor*.

### **Update Preview**

Displays what elements would be affected by an **update** operation, and enables you to easily select the elements and perform operations on them in advance of the update. By default the following columns are displayed:

- Element -- (When "Display of element name..." is set to "single column") The depot-relative path of the element in the workspace, followed by the name of the element as it currently appears in the workspace.
- Name -- (When "Display of element name..." is set to "separate name and in-folder") The name of the element as it currently appears in the workspace.
- In Folder -- (When "Display of element name..." is set to "separate name and in-folder") The depot-relative path of the element in the workspace.
- Status -- See [AccuRev Element Status](#) on page 89.
- Action -- What will happen to the element if you proceed with the update. Possible values are move, remove, create, overlapped, re-link, replace content, and no such element. For elements with a status of no such element, the only operations that are enabled are *History* and *Properties*.
- Real -- The workspace and version where the current version of the selected element is defined.
- New Name -- If the element has been renamed in the backing stream, the name which will appear in the workspace after the update.

Note: The **File->Update Preview** menu option now displays this filter rather than the text display generated in previous releases.

### ***Overlap***

Select elements whose status includes **(overlap)** -- the current version in the parent stream is not an ancestor of this version. This means there might be *content changes* or *namespace changes* in the parent stream version that are not present in this version.

You must perform a *Merge* with the parent stream version before you can *Promote* your version. If you're in a dynamic stream, you must perform the *Merge* in the *Change Palette*.

### ***Deep Overlap***

Select elements that satisfy the **Overlap** search criterion in the current workspace or stream, along with **Overlap** elements in the *parent* stream, in the *grandparent* stream, and so on -- all the way up the depot's stream hierarchy. From the Deep Overlap filter display, you can *merge* your changes to resolve the overlap and then *keep* the merged file. The Overlap Filter display updates to show the merged element with a **(kept) Member** status, and you can select it and *promote* to the backing stream.

### ***Underlap***

Select elements whose status includes **(underlap)** -- the current version in the parent stream is not an ancestor of this version, but the parent-stream version already contains all the changes of this version. In most cases, you can safely *Revert to Backed* your version, so that you'll get the parent-stream version the next time you *Update* the workspace (see [Revert to Backed Dialog](#) on page 128). Alternatively, you can perform a *merge*, just as with **(overlap)**-status elements.

### ***Modified in Default Group***

(workspace only) Select elements whose status includes both **(modified)** and **(member)**. This is the intersection of the sets of files selected by the **Modified** and **Default Group** search criteria.

### ***External***

(workspace only) Select files and directories that exist in the *workspace tree*, but have never been placed under *version control* with the *Add to Depot* command.

### ***Missing***

Selects elements that *should* be present, but aren't. That is, there's a version of the file or directory in the workspace stream, but the file or directory was removed from the *workspace tree* by an operating system command or some non-AccuRev program.

### ***Stranded***

Select the elements in the default group that have become *stranded*. In the Details pane, a stranded element is identified by its element-ID, since it has no pathname in the workspace or stream.

*Notes: Example*

Here's the most typical scenario for stranding a file in a workspace:

1. Create a new version of a file with Keep. This places the file element in the workspace's default group.

2. Remove the file's parent directory with Defunct. With the parent directory gone, there's no pathname in the workspace to the file element.

For more information, see [Version Control of Namespace-Related Changes](#) on page 49 in *AccuRev Technical Notes*.

## Defunct

Selects the elements in the default group of a workspace or dynamic stream whose status is **(defunct)**.

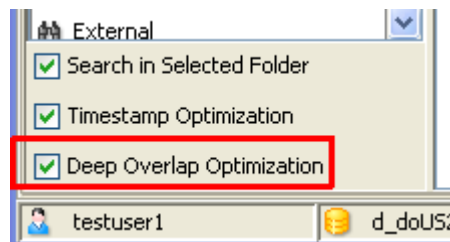
## Search Optimizations

Many of the File Browser's searches described above require that AccuRev consider every file in your workspace, even the *external* objects that you haven't placed under version control (e.g. editor backup files, files produced by software builds). If your workspace contains many thousands of files, such operations can be time-consuming. AccuRev can use optimizations during a full-workspace search to significantly improve search performance, and to reduce "visual clutter":

- [Timestamp Optimization: Controlling the Determination of \(modified\) Status](#) on page 97
- [Pathname Optimization: Selective Processing of External Objects](#) on page 95

## Deep Overlap Optimization

The **Deep Overlap Optimization** checkbox controls whether or not to consider and display elements beyond a time basis stream.



When checked (default), deep overlap elements beyond a time basis stream are not displayed. This improves performance, and also simplifies the display by not showing elements that are not relevant.

When unchecked, these elements are displayed, but with orange highlighting.

Name	In Folder	Overlap Stream	Status	Version
file.cpp	\.\src\	s_doUS200_1_1_1	(overlap)(member)	s_doUS200_1_1_1\1
file.cpp	\.\src\	s_doUS200_1_1_1	(overlap)(member)	s_doUS200_1_1_1\1
file.cpp	\.\src\	s_doUS200_1_1_1_1	(overlap)(member)	s_doUS200_1_1_1_1\1

This setting is not saved between GUI sessions.

## Operating on Elements Selected by a Search

The set of elements located by a search are displayed in the Details pane. The operations that you can perform on these elements are described in [Details Pane Command Reference](#) on page 80.

*Notes:*

- Selecting all the objects located by a search

Selecting all objects in the Details pane can be particularly useful after a search. For example, to promote all files that you've saved with Keep:

1. Click Kept in the Searches pane
  2. Select one file in the details pane
  3. Press Ctrl-A to select all the elements in the Details pane
  4. Invoke the Promote command.
- Disappearing search results

A file can disappear from the Details pane if you change it in a search-related way. For example, after you promote all the files displayed by a Kept search, the files' status changes from (kept) to (backed). Accordingly, the files disappear from the Kept search results.

## File Browser: Searching by Element Name

In addition to searching by element status as described in the previous section ([File Browser: Working in the Searches Pane](#) on page 59), you can also search your Workspace for elements with names that include specified characters.

To enter a search string, type the characters in the Search field at the right end of the toolbar:



Press Enter or click the binocular icon when done. AccuRev displays a “Stat processing” status box while it searches the database for all elements in the Workspace context that include the string that you entered.

Note: Unless you have the **Search in Selected Folder** checkbox selected (see below), the search returns results from the entire Workspace, not just the file structure that is displayed when you initiate the search.

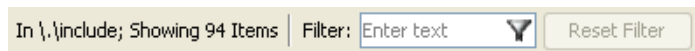
Click the Run in Background button if you want to do other things while AccuRev searches. You can enter any character that is legal for an AccuRev element name. Wildcards ("\*" and "?") are accepted. AccuRev searches the Workspace for all elements that include the string in any part of the name. So for example, performing a search for "test" will return all elements with names like "Testplan.doc", "regression\_test\_suite", "script.test".

Note that:

- The search can be narrowed down to a particular directory by clicking that directory in the tree view and enabling the **Search in Selected Folder** checkbox.
- Search is available only for locally accessible Workspaces, not for Workspaces on a different machine. Search is also not available for dynamic streams.
- If the search results return a directory, you can click on it to view that directory in directory mode. The directory tree on the left expands and selects the correct node/directory.

## Filtering Your Search Results

To fine-tune the results of your search, you can enter characters in the **Filter** field. As you type a string, the search results shown in the File Browser display will be narrowed down to only those containing the characters you enter. The rules for Filter strings are the same as for the search.

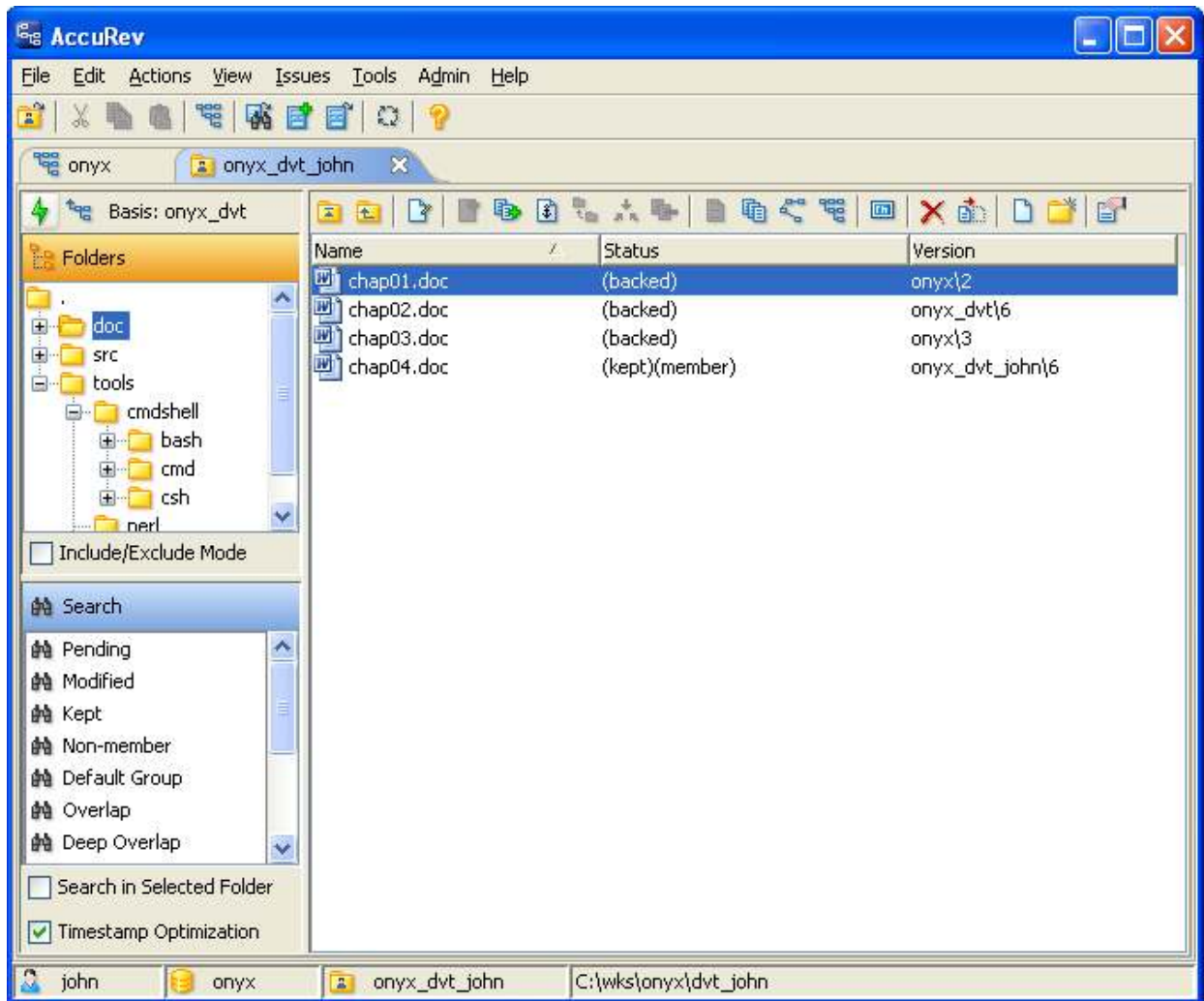
A screenshot of a software interface showing a search filter. On the left, it says 'In \.\\include; Showing 94 Items'. To the right of this is a label 'Filter:' followed by a text input field containing the placeholder text 'Enter text'. To the right of the input field is a small icon of a magnifying glass. Further to the right is a button labeled 'Reset Filter'.

## File Browser: Working in the Details Pane

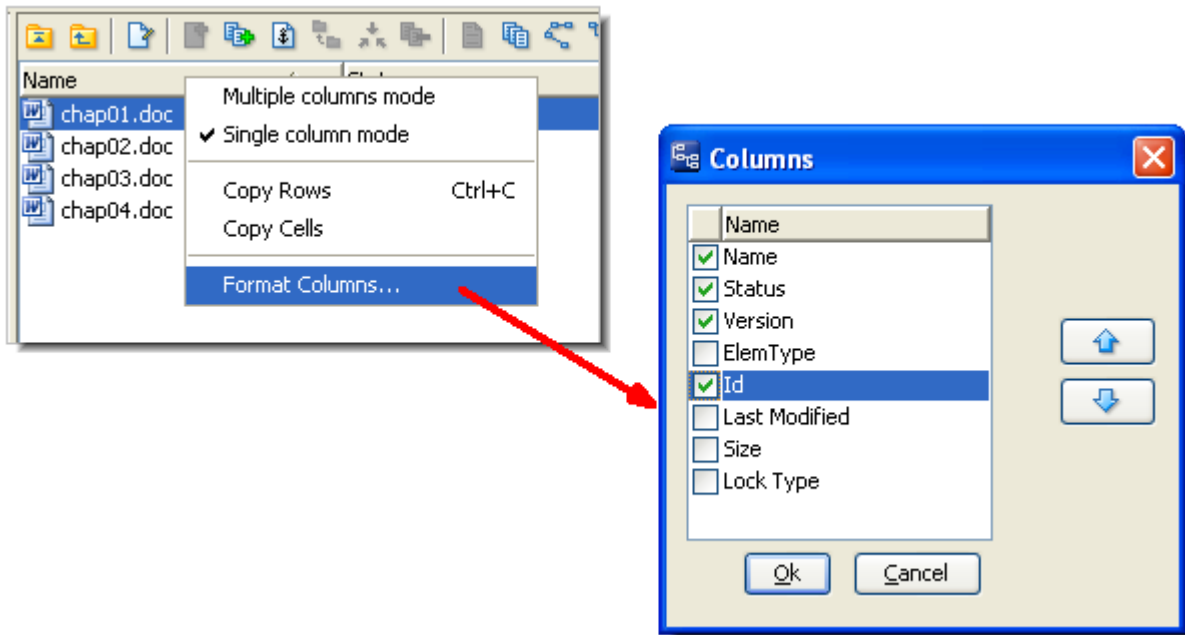
This topic discusses the tasks that you'll probably perform most often with AccuRev -- working with elements in the Details pane of the File Browser.

### Details Pane Layout

The Details pane displays the contents of a particular directory (folder) if you've made a selection in the Folders pane, or the set of objects resulting from a search that you've selected in the Searches pane.



The display includes several columns of information. You can control which columns are displayed: right-click any column header and choose *Format Columns* from the context menu:

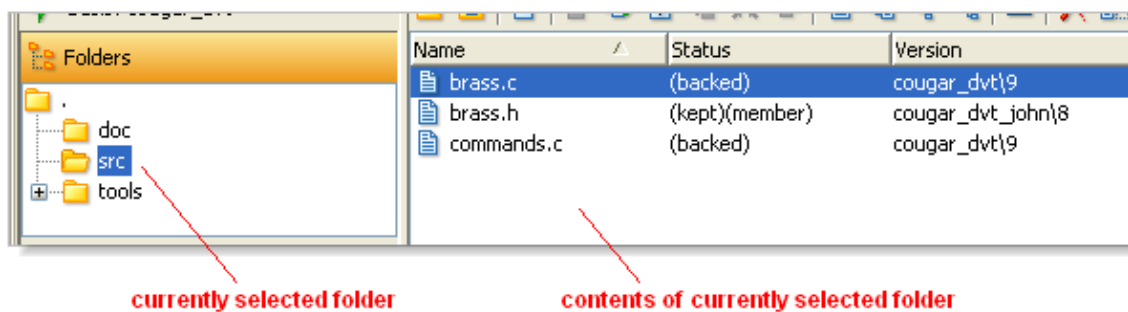


The Details pane columns are:

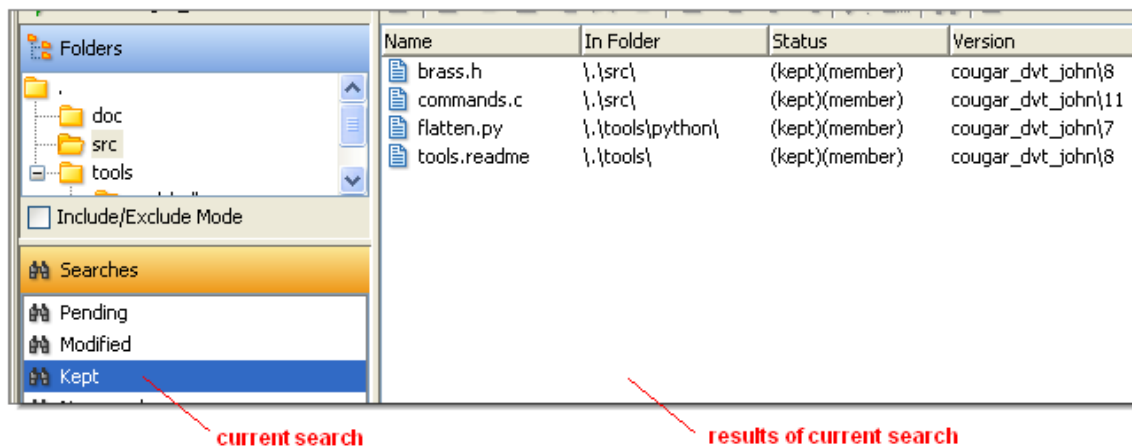
### *Name*

The name of the object, either a file or a directory, in this workspace or stream. AccuRev allows you to rename and relocate objects, so a file might have a different name and/or a different directory location in another workspace or stream.

When the Details pane is displaying the contents of a single folder, the *Name* column shows the name of each object in the folder. The highlight in the Folders pane indicates which folder is being displayed.

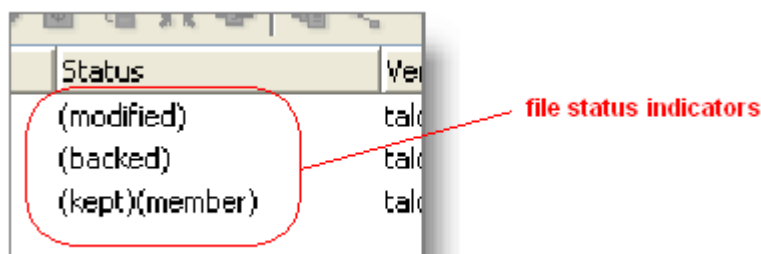


When the Details pane is displaying the results of a *search* through the entire workspace or stream, a *user preference* controls whether element pathnames are displayed as a single column (*Element*) or two columns (*Name* and *In Folder*).



## Status

One or more keywords that indicate the AccuRev *status* of the element in this workspace or stream.



See also:

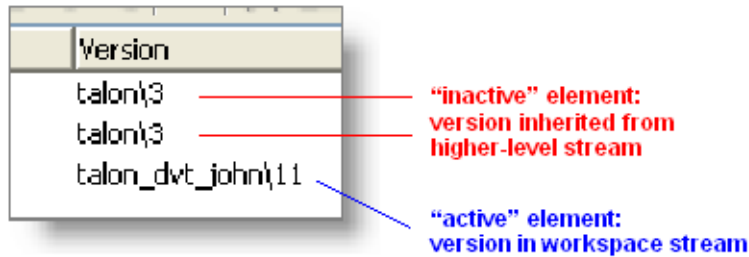
[AccuRev Element Status](#) on page 89

## Version

A *version-ID* that indicates the version of the element that currently appears in this workspace or stream.

- Workspace:** If you have modified an element, but not yet promoted it to the backing stream, the version-ID indicates a version in the *workspace stream*. These are described as "*active* elements", or as "members of the workspace's *default group*". For inactive elements (those you haven't modified or activated with *Anchor* or *Send to Workspace*), the Version column indicates a version from some higher-level stream, which your workspace stream *inherits*.

If this column is blank, the file or directory has not been placed under *version control*. Such objects have the status *external*.



**Stream:** If the element is currently *active* in the stream (is in the stream's *default group*), the version-ID includes the name of this stream. Otherwise, the version-ID includes the name of the higher-level stream from which this stream inherits the version.

### ***ElemType***

File element: the *element type* of this version of the element: **text**, **p<sub>text</sub>**, or **binary**.

Directory element: the keyword **dir**.

Link element: the keyword **link**.

### ***Id***

The *element-ID* of the element.

### ***Last Modified***

A date-timestamp, indicating the last time this element was modified. In a workspace, this is the same as the timestamp reported by the operating system for the object in the *workspace tree*.

### ***Size***

The size of this version of the file (or the file version referenced by this version of an *element link* or a *symbolic link*). The size of a directory is reported as "0KB".

### ***Lock Type***

Either **parallel** or **serial**, indicating the element's current development mode. See [The Locks Command](#) on page 53.

## **Options that Affect the Details Pane Display**

Several options provide fine-tuning over which elements are displayed in the Details pane, and affect which keywords appear in the *Status* column:

- (Workspace only) [File Browser: Controlling the Display of External Objects](#) on page 102
- (Workspace only) [Timestamp Optimization: Controlling the Determination of \(modified\) Status](#) on page 97
- [File Browser: Controlling the Display of Element Names](#) on page 89

Switching in and out of *include/exclude mode* also changes which elements are displayed in the Details pane.

## Performing Operations in the Details Pane

Working with elements in the Details pane follows this pattern:

1. Select one or more files and/or directories.
2. Invoke an AccuRev command, either:
  - ... by clicking one of the File Browser's toolbar buttons, or
  - ... by right-clicking a selected object and choosing a command from the context menu, or
  - ... by choosing a command from the submenu of the GUI window's main menu.

The file-selection gestures are the same ones used by Windows Explorer:

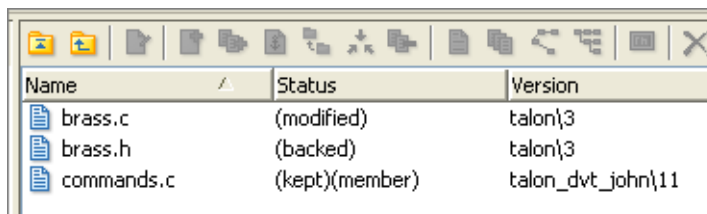
- To select a file, click it with the left mouse button.
- To select a contiguous range of files, click-and-drag with the left mouse button.
- To select all the files in a directory, select any one of them, then press Ctrl-A.
- To add or subtract a file from an existing selection, hold down the Ctrl key and click that file with the left mouse button.
- To extend an existing selection to a particular file, hold down the Shift key and click that file with the left mouse button.
- Typing a character selects the next file or directory whose name begins with that character (if the Details pane listing is sorted on the Name column).

## Common Usage Scenarios


As a version control system, AccuRev keeps track of changes that users make to files and directories. The following sections describe common usage scenarios, showing how AccuRev change-tracking is reflected in the File Browser's Details pane: in changes to the status of objects, and in creation of new versions of objects.

### Editing a File's Contents

Perhaps the most common scenario is "non-conflicting development": modifying a file that no one else is working on concurrently. Initially, the Status column in the Details pane shows the file's status as **(backed)**: your workspace contains an unmodified copy of a version in the backing stream. The Version column contains a version-ID, indicating which backing-stream version it is:

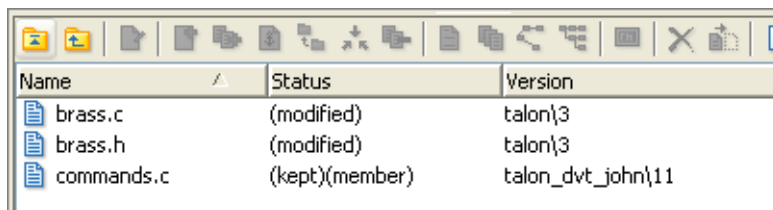


Name	Status	Version
brass.c	(modified)	talon\3
brass.h	(backed)	talon\3
commands.c	(kept)(member)	talon_dvt_john\11


Invoke the  *Edit* command on the file, using the toolbar button or the file's context menu. This launches a text editor session on the file. (You can use environment variable AC\_EDITOR\_GUI or EDITOR to control which text editor is launched.)

When you end the edit session, the file's status changes to **(modified)**. This indicates that you've modified the file in your workspace since the last time you synchronized the file in your

workspace with the depot, for example with a *Keep* or *Update* command. (That is, you've changed the file in the *workspace tree*, but not the element in the *workspace stream*.)

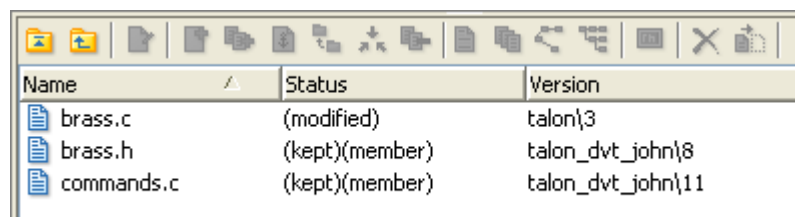


Name	Status	Version
brass.c	(modified)	talon\3
brass.h	(modified)	talon\3
commands.c	(kept)(member)	talon_dvt_john\11

You can edit the file as many times as you wish. The file's status remains **(modified)**. To have AccuRev record a new version of the file, invoke the  *Keep* command on it. If you wish, enter a comment in the dialog box that appears. The comment string becomes a permanent annotation to the version, viewable with the *History Browser* (see [The History Browser](#) on page 187).



The *Keep* command creates a new version of the file in the workspace stream. This version is said to record a *content change* to the file. The File Browser also has commands for performing *namespace changes*.

The file's status changes to **(kept)(member)**; this indicates that you've recorded a new version, and that the file is currently *active* in your workspace (is a member of the workspace's *default group*).





Name	Status	Version
brass.c	(modified)	talon\3
brass.h	(kept)(member)	talon_dvt_john\8
commands.c	(kept)(member)	talon_dvt_john\11


The Version column shows the version-ID of the newly created version. Note that this version is recorded in your private workspace stream (in this example, *talon\_dvt\_john*); previously the Version column indicated that your workspace contained a version from the public backing stream (*talon*).

You can continue modifying the file with the  *Edit* command, and saving new versions in the depot with the  *Keep* command. The file's status will alternate between **(modified)(member)** and **(kept)(member)**. The persistence of the **(member)** indicator reflects the fact that the file remains active in your workspace until you *promote* your changes to the backing stream or *undo* your changes.

## Renaming or Moving a File

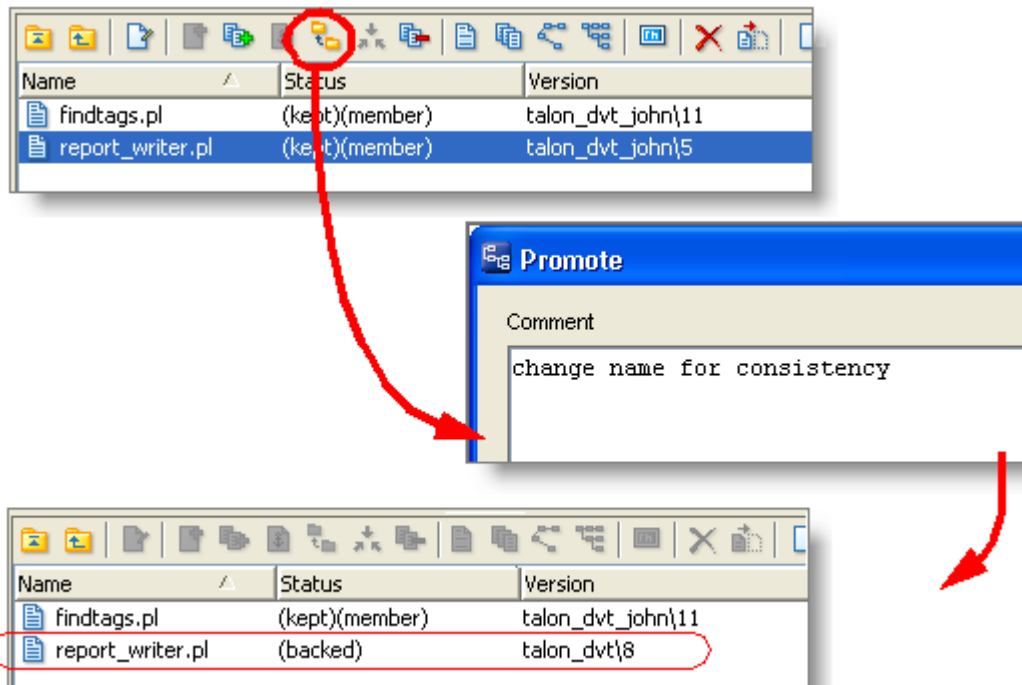
In addition to tracking changes to the contents of files, AccuRev tracks *namespace changes*:

- To rename an element within the same directory, invoke the  *Rename* command. You can't use this command to rename **(external)** objects.
- To move a file to a different directory in the depot, right-click the file and select  *Cut from* the context menu. Then right-click the destination directory in the Folders pane (not the

Details pane) and select  Paste from the context menu. You can cut-and-paste a multiple-file selection in the same way.

You can also use the *Rename* command to move a file to a different directory; specify a relative pathname (such as `..\otherdir\myfile.c`) as the new name for the file.

AccuRev records a namespace change to a file in the same way it records a content change: by creating a new version of the file in the workspace stream.



As with a content change, the file's status changes to **(kept)(member)**. Making a namespace change to a file activates it -- creates a new version in the workspace stream and makes it a member of the workspace's default group -- just like *Keep*'ing a content change.

You can also *rename and/or move a directory*.

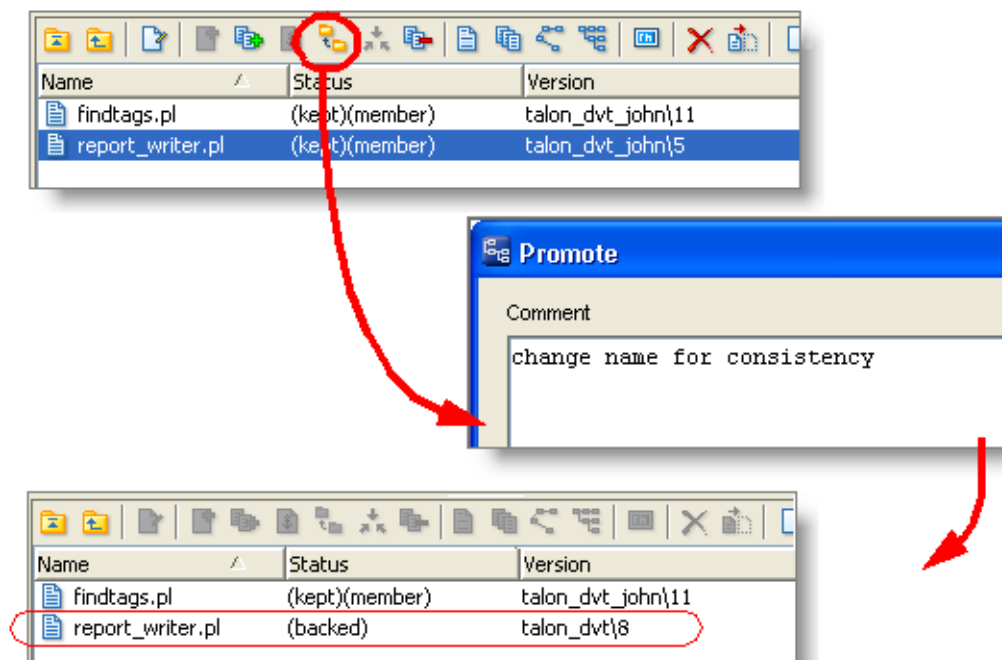
*Notes:*

- Status of a renamed file element: a special case  
If the file also has content changes that have not yet been saved with Keep, the status becomes **(modified)(member)**.
- Reusing the old element name  
After you rename an element, you are free to create a new element with the old name. If you do so, and then you decide to discard your renaming of the original element (Revert to Backed command), then the original element disappears from your workspace. (See also: [Caution on Reusing the Name of a Renamed Element](#) on page 126.)

### Following Through by Promoting the Changes

Initially, the content changes and/or namespace changes you make to a file are recorded only in your workspace's private stream. This keeps your work isolated from your colleagues' work.

When you're ready to share your changes to a file with your colleagues, you *Promote* the active version from your *workspace stream* to the *backing stream*. This makes your changes available to all workspaces that are based on the same backing stream.



Note how the version-ID in the Version column changes:

- Before the promotion, it indicates the particular version of the file that is active in your workspace (in this example, version talon\_dvt\_john\5 ).
- After the promotion, it indicates that version's newly created version-ID in the backing stream (in this example, talon\_dvt\8).

Having been promoted to the backing stream, the file is no longer active in the workspace (is no longer in the default group of the workspace stream). Accordingly, it loses its **(member)** status and returns to being **(backed)** -- its original status before you started working on the file. That is, the workspace returns to "inheriting" the version of the file currently in the backing stream -- which happens to be the version you just promoted there!

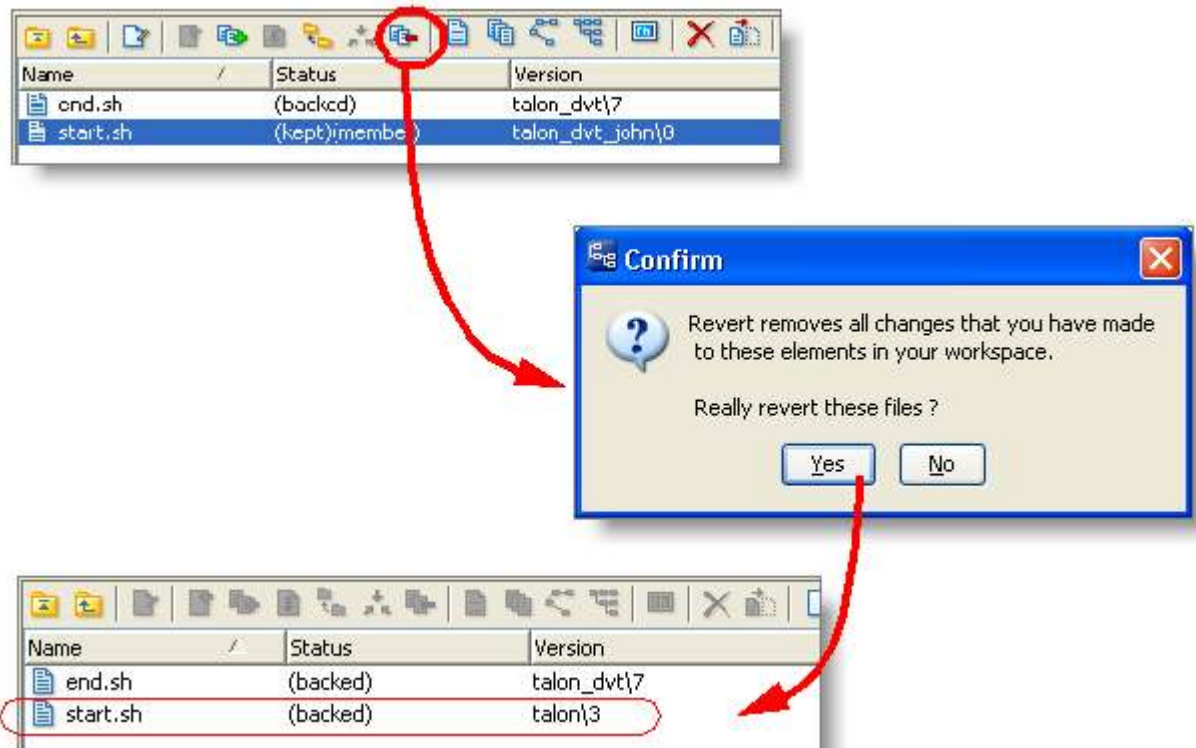
*Note: Combining the Keep and Promote commands*

Strictly speaking, you must *Keep* a file's changes before you can *Promote* them. But as a convenience, the File Browser enables the *Promote* command for files that are **(modified)**, but not **(kept)**. Invoking *Promote* performs both a keep transaction and a promote transaction.

### Following Through by Undoing the Changes

Inevitably, you sometimes decide *not* to share your changes to a file with your colleagues -- instead, you decide to discard the changes altogether. The *Revert to Backed* command undoes all the content and/or namespace changes you've made to an active file. The file's status reverts to **(backed)**, and your workspace "rolls back" to using the version that it contained the last time the file's status was **(backed)**. It might be a version that you brought into your workspace with a

recent *Update* command; or it might be a version that you created in your workspace, then *Promote*'d to the backing stream.



#### Notes:

- Reverting namespace changes  
If your changes to a file included renaming it or moving it to a different directory, invoking Revert to Backed causes the file to disappear from its new location and return to its original pathname.
- Variant command: Revert to Most Recent Version  
The variant command, Revert to Most Recent Version, is useful if you've modified a file's contents repeatedly, creating one or more intermediate versions in your workspace with the Keep command. Revert to Most Recent Version discards any content changes you've made since the most recent Keep. The file's status reverts from **(modified)** to **(kept)**. The file remains active in the workspace, so it retains its **(member)** status.

### Changing a Directory

In addition to tracking changes to files, as discussed in the preceding sections, AccuRev tracks changes to directories. AccuRev's model for directory-level changes is simple, but somewhat different from the model used by the operating system (and by some other version-control systems). AccuRev considers the following to be changes to a directory:

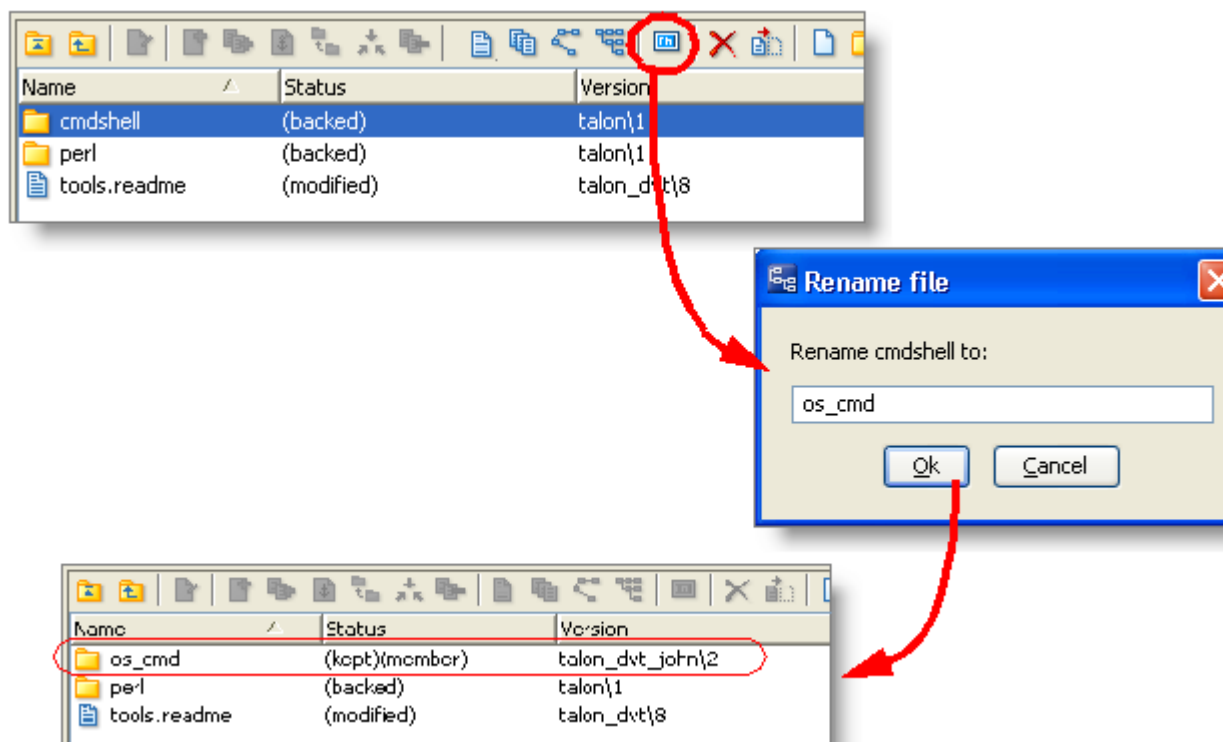
- Renaming a directory
- Moving a directory to another location in the depot's directory hierarchy
- Deleting a directory

The following are *not* changes to a directory:

- Creating a new file (it's a change to the file itself)
- Renaming an existing file (it's a change to the file itself)
- Deleting a file (it's a change to the file itself)

Note that it is only changes involving a directory's pathname that are considered to be changes to the directory itself. Changes to a directory's contents are not considered to be changes to the directory itself.

You change a directory's pathname in the same way you change a file's pathname -- with the *Rename* command or with *Cut* and *Paste* commands. When you make such a change, AccuRev records a new version of the directory in the workspace stream.



The new version-ID appears in the Version column of the Details pane (in this example, *talon\_dvt\_john\2*). And the directory's status changes to **(kept)(member)** -- just as it does for a file's namespace change.

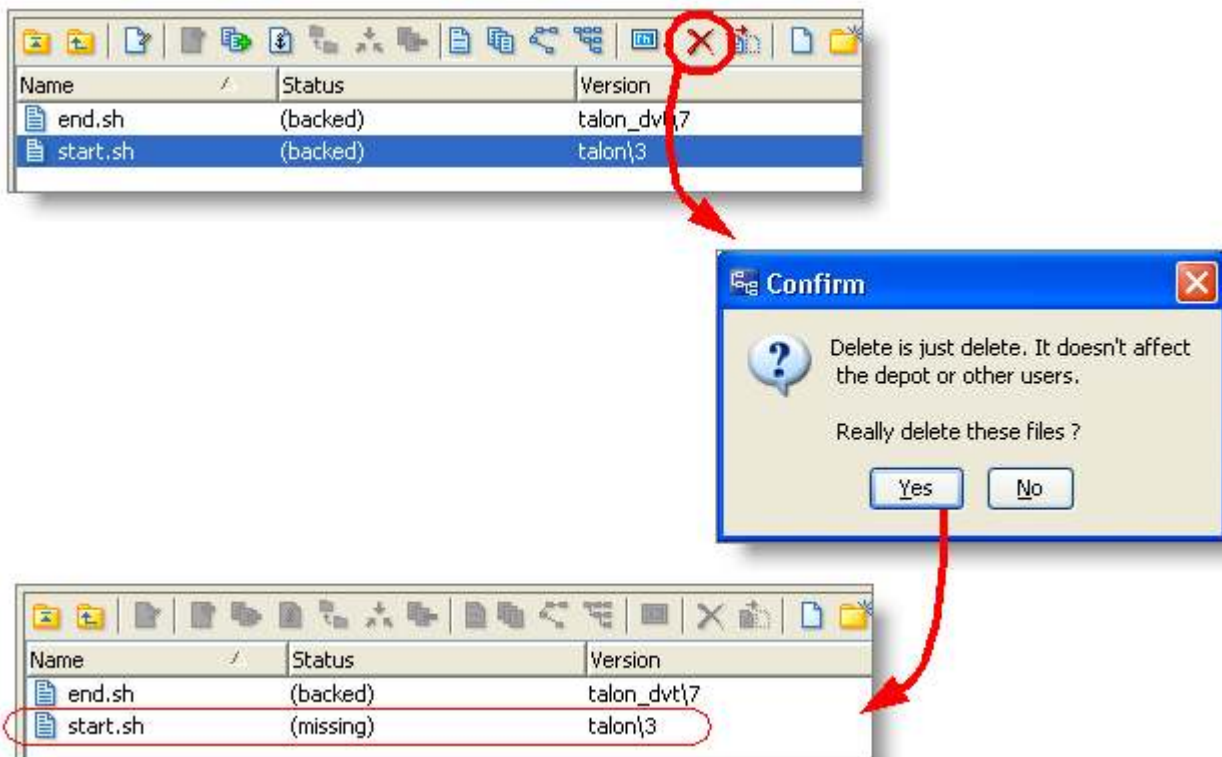
### Deleting a File -- Accidentally or Temporarily

An AccuRev workspace tree is an ordinary directory tree, typically located on your machine's hard drive. Nothing prevents you (or perhaps, some rogue cleanup script) from using operating system commands to delete one or more of the files under version control. On occasion, you may even want to delete some files temporarily -- for example, to test the robustness of your build or installation procedure.

By definition, deleting a file at the operating system level makes it disappear from disk storage. Operating system tools, such as Windows Explorer or the UNIX *ls* command, will detect that the file no longer exists. But the file does *not* disappear from the File Browser display. AccuRev

knows that the file *should* be in the workspace, because the file element still exists in the workspace's built-in stream. (The workspace stream is located in the AccuRev depot. It's unaffected by the operating system's delete-file commands.)

Accordingly, when a version-controlled file is deleted at the operating system level, the File Browser continues to list it, but indicates the file's status as **(missing)**.



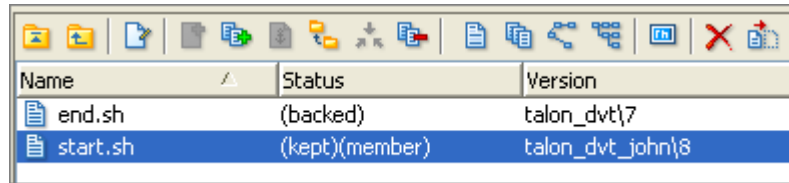
Note that the Version column continues to indicate which version of the file should be in the workspace. To restore that version, invoke the *Populate* command from the missing file's context menu.

Deleting a directory at the operating system level is similar to (but potentially more destructive than) deleting a file. The entire directory subtree is deleted from disk storage, and the directory is listed by the File Browser as **(missing)**. To recover the entire directory tree, invoke the *Populate* command from the directory's context menu. Be sure to select the Recursive option from the *Populate* dialog.

### Deleting a File -- Intentionally and Permanently

Sometimes, you want to delete a file permanently (see note). That is, you want the file to disappear from your workspace, and from other users' workspaces, too. The file might be related to a product feature that was cancelled. Or perhaps a code reorganization rendered the file unnecessary.

Deleting a file at the AccuRev depot level (rather than simply at the operating system level) is called *defuncting*, and is implemented by the *Defunct* command.



Name	Status	Version
end.sh	(backed)	talon_dvt\7
start.sh	(kept)(member)	talon_dvt_john\8

Defuncting a file removes it from the workspace's disk storage (that is, deletes the file at the operating system level). In addition, the *Defunct* command is recorded in the depot. It may be surprising at first, but AccuRev manages the defuncting of a file in the same way as it manages the creation of new versions. To AccuRev, defuncting is just another kind of change that can happen to a file:

- Defuncting "activates" a file in the workspace stream, recording the fact that you've made a change to the file. In addition to getting **(defunct)** status, the file gets the **(kept)** and **(member)** statuses, just as if you had performed a Keep command.
- AccuRev records the change as a new version of the file in the workspace stream (in this example, version talon\_dvt\_john\10).
- At first, the defuncting of a file is isolated to your own workspace. The file continues to exist in other users' workspaces.
- To "share" the defuncting of a file, you Promote the change to the backing stream. This causes the file to disappear from your workspace stream, and from the File Browser display. The file will disappear from other users' workspaces when they invoke the Update command.
- As always, you propagate the change -- in this case, removal of the file -- throughout the depot by promoting the defuncted file from the backing stream to the depot's higher-level streams.

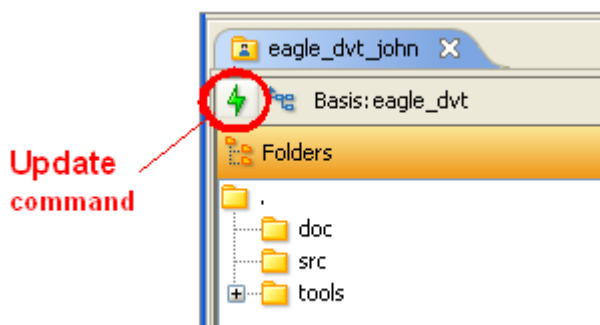
*Note: Defuncting isn't really permanent (and the file isn't even really gone)*

The Defunct command does not actually remove the file element from the AccuRev repository. Because AccuRev is TimeSafe, old versions of the file continue to exist in the repository. You can see such versions in **snapshots** of streams, in **History Browser** displays, etc. You can restore such elements with the CLI command undefunct.

## Incorporating Others' Changes into Your Workspace

As you and your colleagues make changes to a project's code base, the contents of your private workspaces diverge more and more. It's a software-development best practice to periodically incorporate others' changes into your workspace, effectively resynchronizing the workspaces

(partially or fully). The *Update* command performs this task, by copying versions that your colleagues have *Promote*'d to the backing stream into your workspace.



This is an oversimplification. For some elements, you must use the *merge* operation to incorporate other's changes into your workspace. For a complete discussion of the update operation, see [Updating a Workspace](#) on page 28.

### Merging Your Changes with Someone Else's Changes

AccuRev supports concurrent development: two or more users can start with the same version of a file and make changes to that file independently -- both *content changes* and *namespace changes*. After one of the users *Promote*'s his changes to the backing stream, each of the others must *Merge* her own changes with the newest version in the backing stream, using the *Merge* tool. We describe here how to initiate a merge operation.

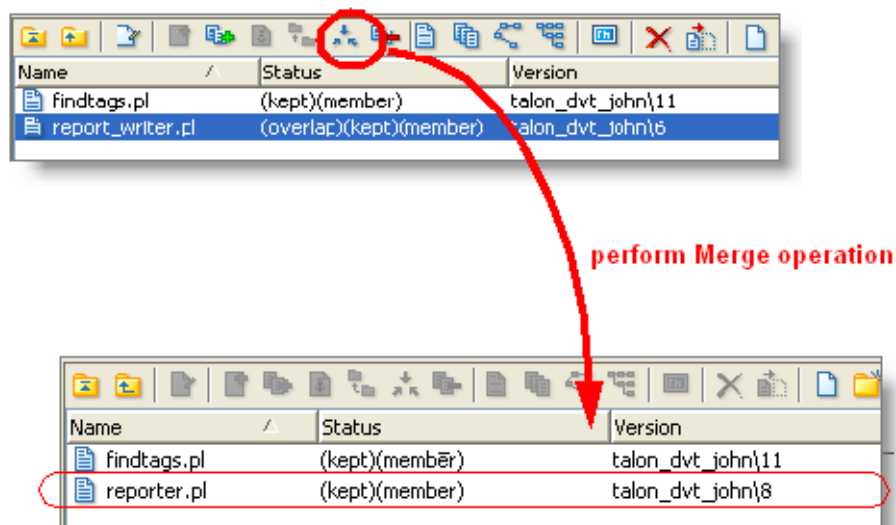
As you work on a file, *Keep*'ing intermediate versions in your workspace, the file's status alternates between **(modified)** and **(kept)**. At any point, you may notice that an additional indicator, **(overlap)**, appears in the Status column. To make sure you notice, the File Browser displays the entry with a yellow highlight.

A screenshot of the AccuRev File Browser showing a table of files. The table has three columns: 'Name', 'Status', and 'Version'. The first row is 'findtags.pl' with status '(kept)(member)' and version 'talon\_dvt\_john\11'. The second row is 'report\_writer.pl' with status '(overlap)(kept)(member)' and version 'talon\_dvt\_john\6'. The second row is highlighted in yellow.

Name	Status	Version
findtags.pl	(kept)(member)	talon_dvt_john\11
report_writer.pl	(overlap)(kept)(member)	talon_dvt_john\6

This means that a new version of the file has been promoted to the backing stream. Typically, one of your colleagues has edited the file and promoted her version to the backing stream. It may also be that someone has promoted a new version of the file to a higher-level stream, and the backing stream dynamically inherits the new version "from on high".

Whenever a file's status is **(overlap)**, the *Merge* command is enabled. The execution of a *Merge* command concludes with the *Keep*'ing of a new version in your workspace.



In this example, the *Merge* involved incorporating a namespace change: the file's name has changed from *report\_writer.pl* to *reporter.pl*. When you *Promote* this merged version to the backing stream, the file's status returns to **(backed)**.

*Note: Underlap status*

The status **(underlap)** is similar to **(overlap)**, in that an element has changed in your workspace and also in the backing stream. With **(underlap)**, the changes in your workspace version have already been promoted to the backing stream (from another workspace, or from a stream elsewhere in the depots stream hierarchy). You can resolve an underlap in the same way as an overlap -- by merging and promoting. But in many cases, it is more appropriate to use the *Revert to Backed* command to undo the changes in your workspace.

## Details Pane Command Reference

You can invoke the following commands from the Details pane toolbar or the context menu of a selected file or directory. Some of the commands operate on a multiple-element selection. Other commands are available from the *Folders pane toolbar*.

Only a subset of these commands are available when the File Browser is in *include/exclude mode*.

### **Go to root**

Have the Details pane display the contents of the depot's (and workspace's) top-level directory.

### **Up one level**

Have the Details pane display the contents of the current directory's parent.

If the current directory is the target of a directory link, this command switches to the target's parent, not to the directory containing the link.

## Open (equivalent to double-click)

File:

(Windows) Run the appropriate command on the file, according to its file type (the Windows file-typing system—"file associations"—does not provide for assigning a file type if the filename has no suffix.)

(UNIX) Open a text editor on the file.

Directory: Have the Details pane display the contents of the directory.

Element link or symbolic link: This command operates on the target of the link.

## View

(text files only) Open a text editor on a temporary copy of the currently selected file. (You can use environment variable `AC_EDITOR_GUI` or `EDITOR` to control which text editor gets invoked. If neither variable is set, AccuRev selects a program provided by the operating system.)

Link to a file element: this command operates on the target of the link.

## Save As

Copy the currently selected file or directory to another location. The entire subtree below a directory is copied.

Element link or symbolic link: this command operates on the target of the link.



## Edit

Open a text editor. (You can use environment variable `AC_EDITOR_GUI` or `EDITOR` to control which text editor gets invoked. If neither variable is set, AccuRev selects a program provided by the operating system.) on the currently selected file. (Select exactly one file before invoking this command.)

The File Browser automatically updates the file's status indicators after an edit session. If you have changed the file's contents, AccuRev automatically includes the **(modified)** status indicator in the Status column.

Note: You can also change the contents of files using operating system commands, scripts, and third-party tools, including integrated development environments (IDEs). In this case, invoke the command View > Refresh or press function key F5 to update the Details pane status indicators.]

Link to a file element: this command operates on the target of the link.



## Add to Depot

(workspace only) Place the selected files/directories under version control. All the selected objects must currently have **(external)** status. For each one, AccuRev creates version 1 in the workspace stream.

Link to a directory: If the **(external)** object is a symbolic link (UNIX/Linux) or a junction point (Windows), this command's default is to create an AccuRev element link, not an AccuRev symbolic link. The target of the link must be a directory element visible in your workspace. If you wish to convert all such directories to AccuRev symbolic links, use the *Add links as symbolic links* checkbox in the Add to Depot dialog box.

Note: Windows junction points are not supported on FAT or FAT32 file systems.

## Keep

(workspace only) Create a new version in the workspace stream for each selected file. All of the selected objects must be file elements; none can have **(external)** status and none can be a directory. A dialog appears, in which you can enter a comment string and select other options.

Typically, you invoke the *Keep* command on files that you've been working on, and thus have **(modified)** status. But this is not a requirement. If you *Keep* a file that you have not modified, a new version is created with identical contents.

The first time you *Keep* a file, it becomes active in the workspace (by being placed in the workspace's *default group*). After that, you can modify and *Keep* the file as many times as you like. The file remains active in the workspace until you run the *Promote* or *Revert to Backed* command on it.

Link to a file element: this command operates on the target of the link.

Note: An element link can never have (modified) status; both creating a link and changing its target give the element link (kept) status. Thus, the element link itself never needs to be Keep'ed.

Special case: [Renaming a Modified File Before Keeping It](#) on page 126

## Anchor

(workspace only) Make the selected files/directories active in the workspace (by placing them in the workspace's *default group*). The files/directories themselves are not modified in the workspace tree. Typically, you anchor a file in your workspace to prevent a subsequent *Update* command from overwriting it with a newer version. (*Update* overwrites inactive files only, not active ones.)

File(s) that you *Anchor* must not currently be the workspace's *default group*. *Anchor* creates a new version in the workspace stream. This new version simply records the fact that the file is officially active in the workspace.

Element or symbolic link: this command operates on the link itself.

*Notes:*

- Anchor creates a virtual version

Other commands that create a new version in a workspace (Keep, Move, Defunct, undefunct (CLI only)) create a **real version**. These commands record an actual change to the element, or to its visibility.

But Anchor creates a **virtual version**, because you haven't made a change to the element -- you've merely reinstated some existing version.

See [Real Versions and Virtual Versions](#) on page 137.

- Anchor-required and exclusive file locking workspaces

In an **anchor-required** or **exclusive file locking** workspace, you must run the Anchor command on a file before modifying its contents. When you anchor a set of elements in an anchor-required or exclusive file locking workspace, a dialog appears if one or more of them are active in a sibling workspace. This enables you to select/deselect individual elements to be anchored. Note that the elements that are active in the sibling workspace are initially deselected. This makes it

easy to avoid the situation where multiple users are working on the same file(s) concurrently. The exclusive file locking feature guarantees that this situation won't occur.

See [The Locks Dialog Box](#) on page 53.

## Promote

Send the active version of the selected files/directories to the *parent stream*. The promoted element(s) become inactive in the workspace or stream (are removed from the *default group*). The status of the elements becomes **(backed)**.

Element or symbolic link: this command operates on the link itself, not on the target of the link.

*Notes:*

- Promote creates a virtual version

The version that Promote creates in the parent stream is a **virtual version**. No data is copied from your machine to the repository by this command; it just creates new entries in the repository database.

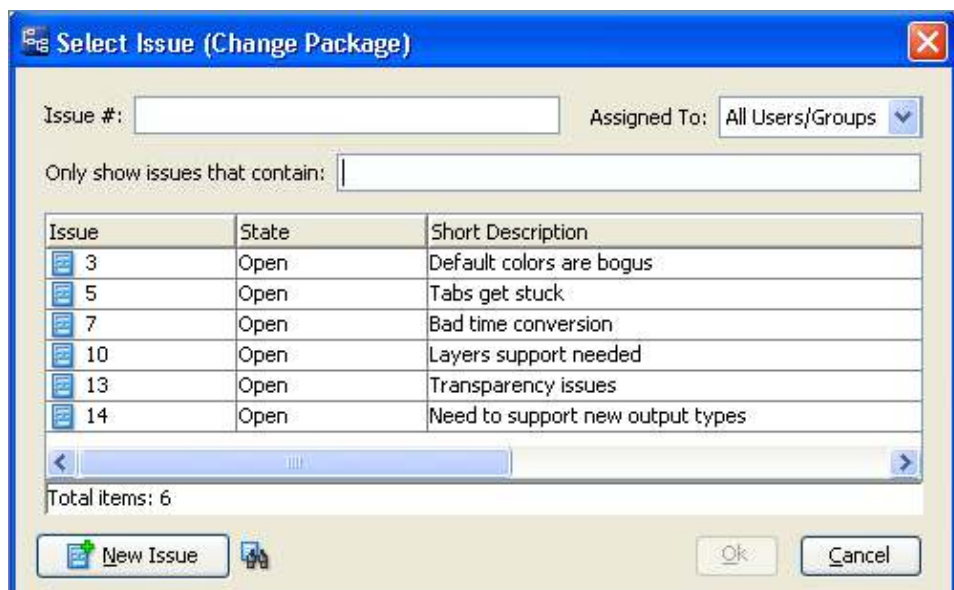
See [Real Versions and Virtual Versions](#) on page 137.

- Promoting from a time-based stream

When you promote a version from a **time-based stream**, it does not become inactive in that stream. Doing so would cause the promoted version to disappear from the stream, because the stream would now inherit a version that was created prior to the **basis time**.

- Integration between configuration management and issue management

If either or both of the integrations between AccuRev's configuration management and issue management facilities is enabled, a particular AccuWork query is executed and its results displayed.



You can select one or more of the issue records and click *Ok*; or you can type one or more numbers (*SPACE*-separated) in the *Issue #* input field.

The integration(s) record information about the *Promote* transaction in the issue record(s) you designate.

### Merge

(workspace only) For each selected file, perform a merge operation, involving two versions: the one in your workspace and the one in the workspace's parent stream. The resulting "merged version" is saved as a new version in the workspace stream.

The merge operation takes into account both the contents of the versions to be merged and the pathnames of those versions. Thus, at the completion of a *Merge* command, a file might have a different name or be located in a different directory, or both.

Element or symbolic link: this command operates on the link itself, resolving conflicting changes to the link object. For example, two users might each have "retargeted" a link (to different pathnames), using **Copy / Paste Link**. Or two users might each have renamed a link (to different names).

### Revert to Backed

For each selected element, remove all changes you've made since the last time you *Promote*'d it, or since the last *Update* of the workspace -- whichever is more recent. For files, this includes both content changes and namespace changes. For directories, this includes namespace changes only. For links, this command operates on the link itself, "undo"ing any changes you've made to the specification of the *link target*. For example, you might have changed the link with the **defunct** or **rename** commands. Or you might have used **Copy / Paste Link** to point the link at a different target directory.

The selected elements become *inactive* in the workspace (they are removed from the workspace's *default group*). The status of the elements becomes **(backed)**.

When you invoke this command in a dynamic stream, the depot's *pre-promote-trig trigger* fires. That's because, like a *Promote*, the command changes which version of the element appears in the stream.

### Revert to Most Recent Version

(workspace only; file element with **(modified)** status only) Replace the file with the most recent version you created with *Keep*. (This command does not undo any namespace changes you made with *Rename* or with *Cut-and-Paste*.) Use this command when you've saved one or more intermediate versions of the file(s), and you want to discard further changes you've made since a *Keep*.

The selected elements remain active in the workspace. That is, they are *not* removed from the workspace's default group. The **(modified)** status of the elements changes to **(kept)**.

If you've modified a file but not yet performed a *Keep* on it, this command works like *Revert to Backed*.

Link and directory elements: this command is disabled, because a link or directory element cannot have **(modified)** status.

### Annotate

(text files only) Open an *Annotate* tab, displaying the contents of the selected version. Each text line is annotated with information on how it was created.

Link to a file element: this command operates on the target of the link.

### **Show History**

Open a *History Browser* tab, containing the transactions involving the selected element.

### **Browse Versions**

Open a *Version Browser* tab, showing all the versions of the selected element, and their interrelationships (ancestry).

### **Browse Stream Versions**

Open a *Stream Version Browser* tab for the selected element. This tab displays the depot's stream hierarchy, much like the *Stream Browser*. But on this tab, each stream represents the *version* of the selected element that appears in that stream.

### **Diff Against Most Recent Version**

(workspace only) Compares the file in your workspace tree with the version currently in your *workspace stream*. Use this command if you've modified the file since the last time you performed a *Keep* on it, or since you last *Update*'d it.

You can invoke this command on a multiple-file selection. A separate tab opens for each file.

### **Diff Against Backed Version**

Compares your version with the version currently in the backing stream.

In a workspace, "your version" is the file in your workspace tree. If the file has **(modified)** status, there is no official version in the repository with the same contents. In a stream, "your version" is that stream's current version.

Note: If you are in a time-based stream, AccuRev uses the version that was in the backing stream at the basis time, not the version currently in the backing stream.

This might include the changes stored in several intermediate versions that you've created with *Keep*. For example, you might use this command to see all the changes you've made to a file since you *Update*'d your workspace and starting working on the file.

Note: This example assumes no one else has promoted a new version to the backing stream in the meantime.]

You can invoke this command on a multiple-file selection. A separate tab opens for each file.

### **Diff Against Basis Version**

(workspace only) Compares your version with the version that you started working with, before making your *recent changes*.

You can invoke this command on a multiple-file selection. A separate tab opens for each file.

## **WIP (Work in Progress)**

Open a *WIP* tab, displaying the active elements in workspaces and/or streams related to the selected stream.

### **Rename**

(workspace only) Change the name of the selected element, using the *Rename* dialog. The new name can be in a different directory within the depot. Thus, this command can perform a "move" as well as a "rename". See [Resolving Namespace Conflicts](#) on page 247.

This command activates the object (by including it in the workspace's default group). The **(kept)** and **(member)** flags are added to its status.

If you change the contents of a file, then rename it without first performing a *Keep*, the version created by *Rename* does not contain the content changes, just the name change. The file's new status reflects this, having the **(modified)** and **(member)** flags.

### **Copy**

(workspace only) Designate the selected object (file, directory, or link) to be the target of an *element link* or a *symbolic link*. Use the *Paste Link* command to complete the creation of the link object.

### **Cut**

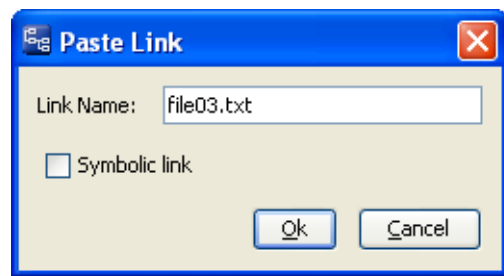
(workspace only) Mark the currently selected element to be moved to another directory in the same depot. To finish the relocation, right-click the destination directory (in either the Folders or Details pane) and select *Paste*.

### **Paste**

(workspace only) Specify the destination for an element that has been marked for relocation with the *Cut* command.

### **Paste Link**

(directory in workspace only) Create an element link or *symbolic link* in the selected directory. The target of the link is the file, directory, or link element on which you previously invoked the *Copy* command.



Note: You can create a link that points to another link. A chain of links can be arbitrarily long, but must end at some file element or directory element.]

By default, the link object has the same name as the target object. But you can change the name in the Paste Link dialog.

If the name you specify does not currently exist, AccuRev creates a new link element (**add** transaction). If the name you specify is an existing link element, AccuRev changes the target of that link element (**keep** transaction).

## Delete

(workspace only) Remove the selected element(s) from the workspace tree . This command does not affect the depot -- AccuRev still thinks the deleted objects should be there, so it continues to list them, with (**missing**) status.

To remove an element from the depot's stream hierarchy, so that its removal will be reflected in other users' workspaces (after you *Promote* the change and they perform an *Update*), use the *Defunct* command instead of *Delete*.

## Defunct

(workspace only) Remove the selected element(s) from the workspace tree , and also change their status to (**defunct**). This also activates the elements in the workspace stream - they're "in the act of being deleted". Accordingly, the elements also get the (**kept**) and (**member**) status flags.

*Notes:*

- Promoting a defunct element

When you *Promote* a defunct element, it disappears entirely from the workspace stream, and from the File Browser display. The element becomes (**defunct**), and also *active*, in the backing stream.

- Updating a defunct element

When you *Update* a workspace below a stream with a (**defunct**) file or link, the file or link is removed from the *workspace tree*. For a (**defunct**) directory, *Update* removes the directory and its entire subtree from the workspace tree.

## Populate

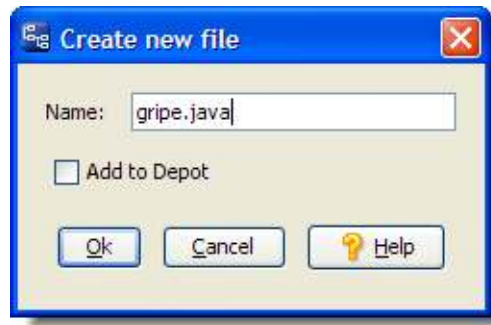
For each selected element, copies the version that is currently in the *workspace stream* to the workspace tree . This has the effect of undoing an accidental deletion, or discarding edits that you have not yet preserved with *Keep*.



A dialog controls the ability to populate entire directory subtrees, along with the ability to overwrite files in the workspace tree.

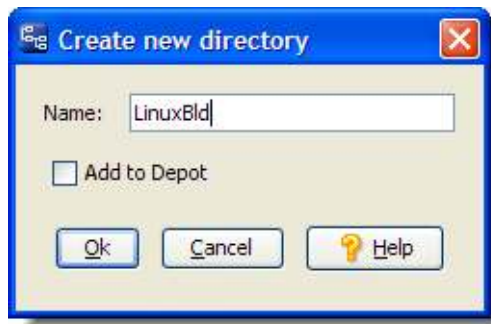
## New File

Bring up a dialog to create an empty file in the current directory, and optionally place it under version control.



### **New Folder**

Bring up a dialog to create an empty subdirectory in the current directory, and optionally place it under version control.



### **Send to Issue**

Create a new entry (or modify an existing entry) in the *change package* section (Changes tab) of one or more *issue records*. The version you specify becomes the *head version* of the change package entry; AccuRev determines the corresponding *basis version* automatically.

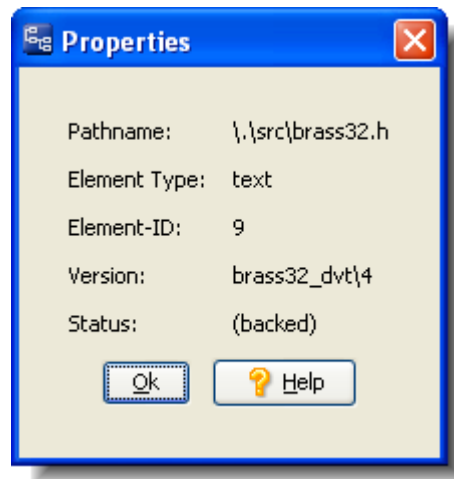
The *default query* for the current depot is executed, and the results are displayed in a dialog. You are prompted to choose one or more of the issue records selected by the query. You can also create a new issue record, whose number will be entered in the dialog.

### **Send to Change Palette**

(dynamic stream only) Load the selected versions into the *Change Palette*, so that they can be promoted to another stream. For some versions, a *merge* might be required prior to the promotion.

### **Properties**

Displays information about the selected element. The data items displayed vary with the type of element.



## File Browser: Controlling the Display of Element Names

Typically, the elements found by a search are located in a number of different folders, throughout the workspace. A user preference controls how to display the locations of the elements:

- Display the complete **depot-relative pathname** of the object as a single string (Element column).
- Display the object's simple name (Name column) separately from the pathname to its parent directory (In Folder column).

Element	Status	Version
\tools\tools.readme	(kept)(member)	brass_dvt_john\11
\tools\perl\reporter.pl	(kept)(member)	brass_dvt_john\10
\doc\chap02.doc	(kept)(member)	brass_dvt_john\13
\src\brass.h	(kept)(member)	brass_dvt_john\8

Use the *Tools > Preferences* command to make this choice ("*Display of element name in tables*"). It affects all instances of the File Browser, along with other tools that display element pathnames.

## AccuRev Element Status

Development work involves making changes to elements -- both files and directories. As files get edited in workspaces, new versions get created in workspace streams, and existing versions get promoted to higher-level streams, AccuRev shows you the current status of each element. Keep in mind that the status of an element in your workspace may be different from its status in other workspaces and streams.

An element's **file status** is indicated by a set of one or more status indicators (or flags), each in the form of a parenthesized word. For example:

(overlap) (kept) (member)

Even though we use the term "file status", each directory element also has a status in each workspace and stream. A subset of the status indicators is used to report directory statuses.

These controls affect the File Browser's display of elements and their file statuses:


- Show External Elements (user preference)
- Timestamp Optimization (File Browser checkbox)

## File Status Indicators


The following sections detail the AccuRev status indicators, organized by category:

### Link-Related Indicators

#### (elink)

The element is an element link. (The link's icon is )

#### (slink)

The element is a symbolic link. (The link's icon is )

#### (missing-target)

For an element link, the target element is not present in the workspace or stream. This can occur if the target element is removed from the workspace tree by an operating system command. It can also result from an *Include from Stream* or *Exclude* command. For a symbolic link, there is no object at the target pathname.

#### (modified-target)

For an element link, the target element has been modified (either a content change or a namespace change) in the workspace or stream.

#### (corrupted)

For an element link in a workspace, AccuRev and the operating system disagree on the link target. That is, the target element recorded in the AccuRev repository differs from the target in the operating system's instantiation of the link in the workspace tree. This can occur if you modify or replace a link using operating system commands instead of AccuRev commands.

A cross-linked element (see **(xlinked)** below) gets **(corrupted)** status if AccuRev does not overwrite the element during execution of the *Include from Stream* command, because the element has **(modified)** or **(kept)** status in the workspace. This should not occur during normal operation.

### Presence of the Element in the Workspace or Stream

#### (defunct)

The element has been marked for removal from the workspace or stream with the *Defunct* command.

*Workspace*: the element has already been removed from the workspace tree (local disk storage). It will be removed from the workspace stream (in the depot) when you *Promote* the element to the parent stream.

*Dynamic stream*: the element will be removed from the stream (in the depot) when you *Promote* the element to the parent stream.

**(external)**

(workspace only) The file or directory has not been placed under *version control*. (There's an object in the *workspace tree*, but no *element* has been created in the *workspace stream*.)

**(excluded)**

The element does not appear in the workspace because it has been excluded, using the *include/exclude facility*. For file elements, it's likely that the exclusion was explicitly set on the directory in which the file resides, or on a higher-level directory that includes the file.

**(xlinked)**

This version of the element appears in the workspace or stream by virtue of a *cross-link* (*Include from Stream* command) — either on the element itself or on its parent directory or a higher-level directory.

**(missing)**

(workspace only) The workspace "should" include a version of this element, but doesn't. This occurs when you delete version-controlled files from the workspace tree using operating system commands, or using the AccuRev *Delete* command. If an operation causes the target of an *element link* to be removed from a workspace, AccuRev removes the element link, also, causing it to have **(missing)** status.

**(no such elem)**

This status can appear under two circumstances:

*Update preview*: The element does not currently exist in your workspace, but will be created if you proceed with the update.

*Element security*: An AccuRev administrator has changed your access to an element or namespace with the **eacl** command during your session, and you are now denied access to something that was already displayed in your File Browser.

**(twin)**

The element is one of multiple elements in the workspace that exist at the same pathname. At most one of these elements can be accessed through the pathname; the other(s) can be accessed through their unique *element-IDs*.

**(stranded)**

The element is active in the workspace or stream, but there currently is no pathname to the element.

See also ...

"Version Control of Namespace-Related Changes" in *AccuRev Technical Notes*.

## **Changes to the Element**

**(modified)**

(workspace only) The file has been modified in the workspace since the most recent *Update* or *Keep*. You can control how AccuRev determines whether a file has this status: see [Timestamp Optimization: Controlling the Determination of \(modified\) Status](#) on page 97.

**(kept)**

A new version of the element has been created with *Keep*, *Rename*, or *Defunct* (or the CLI command *undefunct*) and the file has not subsequently been modified, promoted to the backing stream, or purged (*Revert to Backed*).

**(member)**

The element is *active* in the workspace. Commands that create a new version in the workspace or stream also make the element active there (if it isn't already).

## Relationship to the Version in the Parent Stream

**(backed)**

The version in the workspace stream is the same as the version in the backing stream. And you have not changed the element since the last time you *Promote*'d it or purged it with *Revert to Backed*, or since the most recent *Update* of your workspace.

**(stale)**

(workspace only) The element needs to be updated, because the version in the backing stream has changed since the workspace's latest *Update*. And since you have not changed the element in your workspace, it can be safely updated (no unpreserved changes will be overwritten).

**(overlap)**

The element has changed both in this workspace or stream, and in the parent stream. This indicates that a *merge* is required before you can *promote* your changes to the parent stream. Elements with this status appear with a yellow highlight.

**(underlap)**

The element has changed both in this workspace or stream, and in the parent stream -- but the parent stream already contains the changes in this workspace or stream. In many cases, it is most appropriate to use [Revert to Backed](#) to remove your changes from this workspace stream. In other cases, you can merge and *promote*, just as with an **(overlap)**-status element.

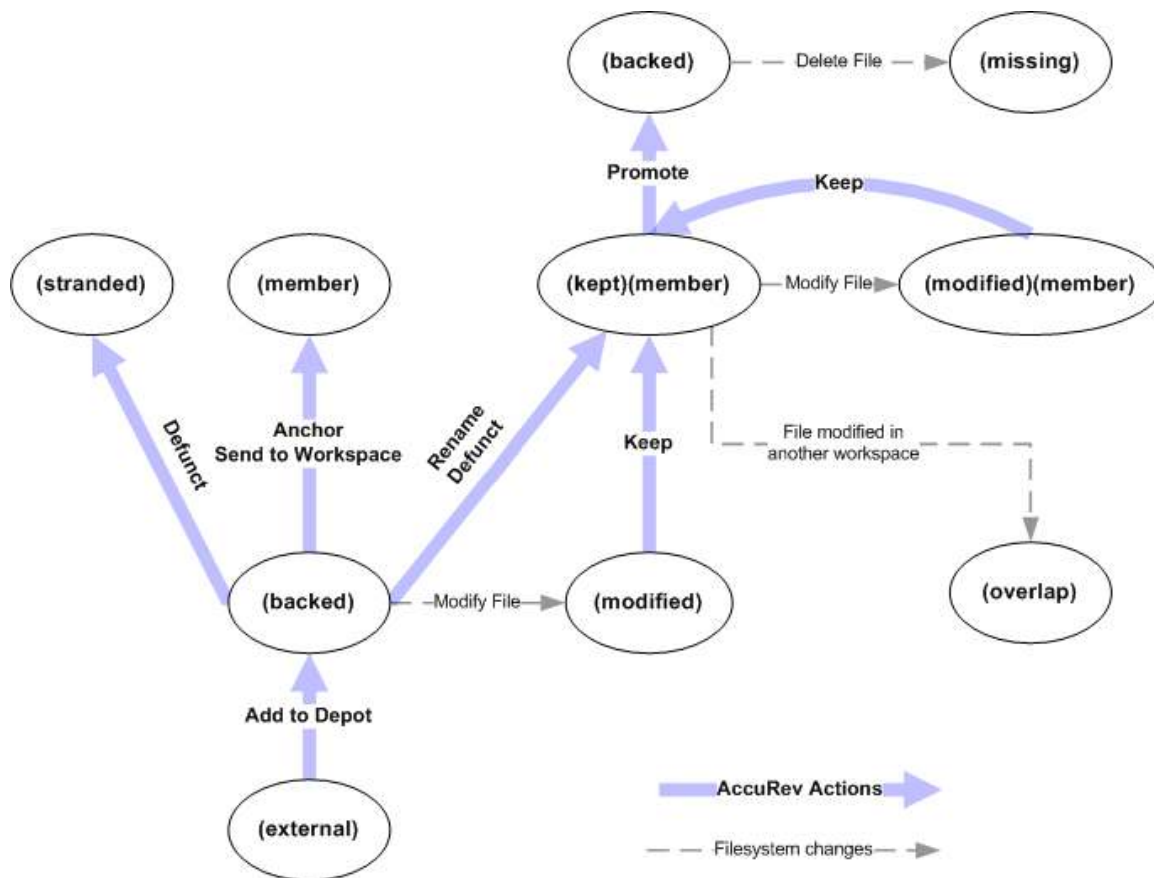
Note: In a workspace, Revert to Backed restores the version of an element that was in the parent stream at the time of your workspace's most recent Update. You might need to perform another Update to bring the version causing the (underlap) status into the workspace.

## File Status Diagram

This diagram shows many of the file statuses, illustrating how an element's changing status indicates its development progress. This diagram suggests that as more work is performed on an element, it "rises to a higher level" of development. (That's the idea behind having a command named "promote".) The diagram also depicts the fact that AccuRev tracks changes to an element in two "dimensions":

- It records changes made by AccuRev commands as new versions in the depot.

- It detects that a file's contents have changed at the operating system level.



This diagram captures the following facts:

- Before you start working on an element, it's unchanged along both the AccuRev dimension and the operating-system dimension. Its status is **(backed)**.
- When you modify the contents of a file, it changes along the operating-system dimension only, and becomes **(modified)**.
- The following involve changes along the AccuRev dimension. This makes the element active in the workspace, so its status indicates that it is a **(member)** of the default group.
- When you Keep a file that you've modified, its status becomes **(kept)**.
- An element can also achieve **(kept)** status through a namespace change -- Rename or Defunct command. This is a change along both the AccuRev and operating-system dimensions.

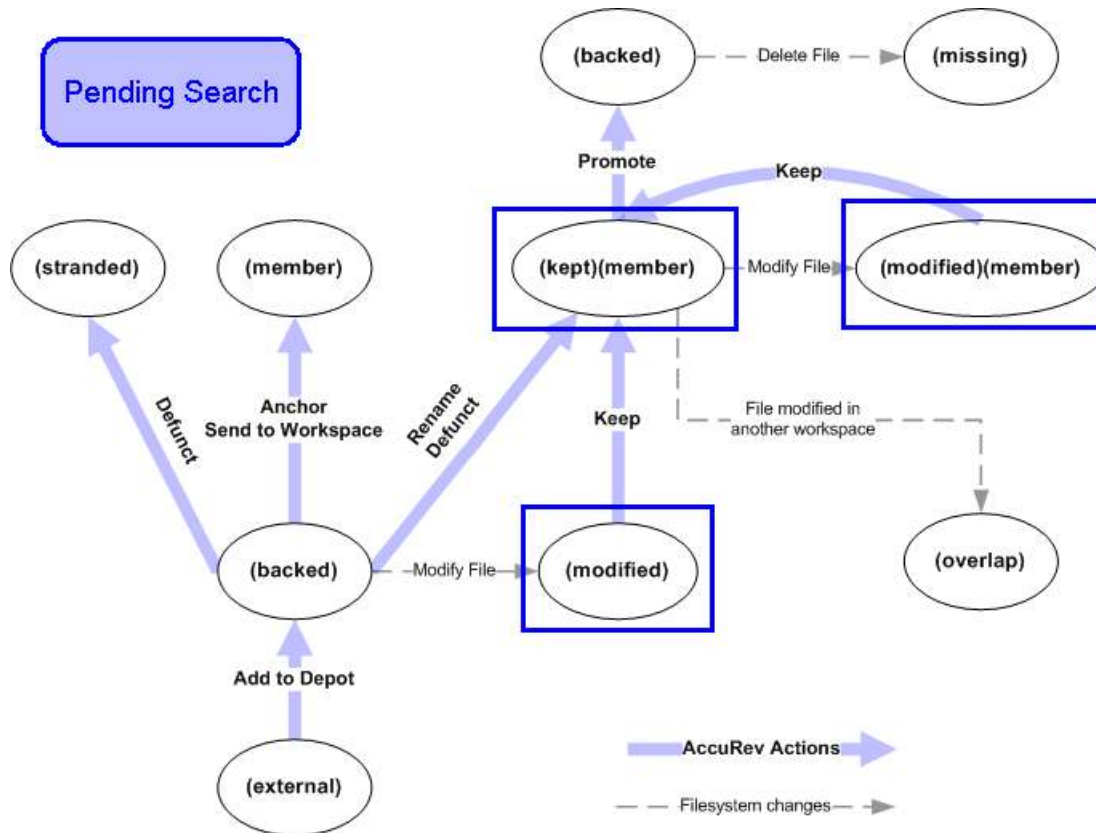
Note: The CLI command undefunct, which reinstates a previously defuncted object, works similarly. This command is not in the GUI.

- The Anchor command change a file along the AccuRev dimension only. The file's status becomes **(member)**.
- You can create any number of intermediate versions of a file in the workspace, by repeatedly modifying the file then Keep'ing it. Throughout this process, the file remains a **(member)** of the workspace's default group. In addition, its status toggles between **(kept)** and **(modified)**.

- You can Promote an element whose status is **(kept)**. This returns the element's status to **(backed)** -- the workspace now uses the newly promoted version in the backing stream.

The following images show which statuses are matched by a particular File Browser search:.

- Pending



- Modified
- Kept
- Non-Member
- Default Group
- Modified in Default Group
- External
- Overlap
- Stranded
- Missing

# Pathname Optimization:

## Selective Processing of External Objects

Note: This topic applies to a File Browser displaying the contents of a **workspace** -- not a **dynamic stream** or **snapshot**.

The File Browser's search facility is very powerful and useful, but searching through an entire workspace or stream can be time-consuming. For example, if your **workspace tree** contains many hundreds or thousands of files, you must wait while all the names are transmitted to the server machine, the AccuRev Server process determines the status of each file, and information on the matching files is returned to the client machine.

This large list of files to be processed may contain a significant number of "don't care" files. For example, a search for **external** files is probably intended to locate source files (with suffixes like **.c** or **.cc** or **.java** or **.bas** ) that you've forgotten to place under version control. You probably don't care about files with suffixes like **.exe** (executables built in the source directory), **.bak** (editor backup files), **.msg** (copies of mail messages, and so on -- because you don't intend to place them under version control.

You can use the environment variable **ACCUREV\_IGNORE\_ELEMS** to specify one or more patterns -- or even individual filenames/pathnames. When it executes certain searches, the File Browser ignores all external files that match this specification. For example, setting **ACCUREV\_IGNORE\_ELEMS** to the following value causes all **.exe** and **.bak** files to be ignored:

**\*.exe \*.bak**

*Note: Controlling the display of external objects*

If you intend to use **ACCUREV\_IGNORE\_ELEMS** to have AccuRev ignore some external objects, you'll probably want to set the preference that enables the display of external objects in the File Browser. But this is not required; you can use **ACCUREV\_IGNORE\_ELEMS** to improve performance, but still suppress external objects from the File Browser display to save visual clutter.

### Eligible Searches

The following File Browser searches use the value of **ACCUREV\_IGNORE\_ELEMS** to filter the names of external objects:

- External
- Modified
- Non-member
- Pending

The File Browser's *Update* command can use the value of **ACCUREV\_IGNORE\_ELEMS** to filter the names of both external objects and elements. It does so only if the user preference [Use Ignore Element Optimization](#) is set.

**CAUTION:** Make sure that the **ACCUREV\_IGNORE\_ELEMS** value matches pathnames of external objects only. If the pathname of a file element is matched during an Update, and that element has (modified) status, AccuRev will proceed with the Update, invoking the routine that overwrites the file with the backing-stream version. This routine will get an error when performing a CRC check on the file.

### Values for **ACCUREV\_IGNORE\_ELEMS**

The value of the ACCUREV\_IGNORE\_ELEMS environment variable must be a SPACE-separated list of filenames, pathnames, and wildcard patterns. You can use either or both of the "standard" wildcards:

- ? matches any one character
- \* matches any sequence of characters (including a zero-length sequence)

Be careful -- the asterisk (\*) wildcard works a bit differently here than in standard UNIX and Windows command processors. It matches any number of characters, including the directory separator (/ or \). With standard UNIX and Windows command processors, the scope of \* is restricted to a single pathname component.

### **Examples:**

**\*.exe**

Matches files whose names end with ".exe", located in any directory.

**\*.exe \*.doc**

**manuals/\*.doc**

Matches files whose names end with ".doc", located in top-level directory "manuals".

Matches files whose names end with ".exe" or ".doc", located in any directory.

**\*/manuals/\*.doc**

**\*.doc README.html**

Matches files whose names end with ".doc", located in any directory. Also matches a file named "README.html" in the top-level directory.

Matches files whose names end with ".doc", located in any directory named "manuals".

For example, assume the following set of manuals:

- (1) manuals/chap01.doc
- (2) manuals/usergd/chap01.doc
- (3) widgetproj/src/manuals/usergd/chap01.doc

Here is how various wildcards will work:

- manuals/\*.doc —Matches (1) and (2), but not (3).
- \*/manuals/\*.doc —Matches (3), but not (1) or (2).
- \*manuals/\*.doc —Matches all three pathnames.

### **Specifying Directories and Their Contents**

A typical application of ACCUREV\_IGNORE\_ELEMS is to have the External search ignore temporary build directories. That is, you want the listing to exclude both the directories themselves and all the files within those directories. If the build directories are named *build\_001*, *build\_002*, etc., you might be tempted to use this pattern:

**\*/build\_??\*/\***

But this pattern matches only the *contents* of the directories, not the directories themselves. Instead, use the following value for ACCUREV\_IGNORE\_ELEMS:

**\*/build\_??? \*/build\_??\*/\***

Don't use the single pattern *\*/build\_??\**. It would match both directories and their contents. But it also might coincidentally match names of some source files, such as *lib/build\_end.c*.

## Notes on Setting the Value

As with any environment variable, the value of ACCUREV\_IGNORE\_ELEMS must be set in the environment before the AccuRev GUI process is started. Here are some guidelines:

### Setting the value on a UNIX/Linux system

When setting the ACCUREV\_IGNORE\_ELEMS environment variable on a UNIX or Linux system, be sure to single-quote or double-quote the value, in order to protect any wildcard characters from being expanded by the shell:

```
export ACCUREV_IGNORE_ELEMS="*.exe *.doc" (Bourne shell family)
setenv ACCUREV_IGNORE_ELEMS "/*.exe *.doc" (C shell family)
```

To determine the current value of ACCUREV\_IGNORE\_ELEMS, use either of these commands:

```
env | grep ACCUREV_IGNORE_ELEMS (or a shorter `grep' pattern)
echo "$ACCUREV_IGNORE_ELEMS" (don't forget the quotes!)
```

### Setting the value on a Windows system

On a Windows system, you can set the ACCUREV\_IGNORE\_ELEMS environment variable in the System applet (on the Control Panel). Alternatively, use the *set* command in a Command Prompt window:

```
set ACCUREV_IGNORE_ELEMS=*.exe *.doc (no quotes!)
```

Don't use quote characters, even if the value includes *SPACES*.

To determine the current value of ACCUREV\_IGNORE\_ELEMS in a Command Prompt window, use either of these commands:

```
set
echo %ACCUREV_IGNORE_ELEMS%
```

## Timestamp Optimization: Controlling the Determination of (modified) Status

Note: This topic applies to *workspaces* -- not a *dynamic streams* or *snapshots*.

In various situations, AccuRev searches some or all of your *workspace tree*, to determine which files's status should include the **(modified)** indicator.

As of AccuRev Version 5.4, there are two different Timestamp Optimization (TSO) mechanisms:

- Traditional, server-based TSO
- Local, client-based TSO

The AccuRev administrator can specify which method is the site default by setting an environment variable as described below in [Setting and Overriding Default TSO Behavior](#) on page 101.

## Traditional, Server-Based Timestamp Optimization

The traditional method of evaluating whether an element in a workspace is modified is based upon comparing the timestamps and checksums of elements in the local **workspace tree** against checksums and a time threshold for the **workspace stream**, requiring communication with the Accurev server:

- When displaying the contents of a directory (folder), the checksum of each file element in that directory must be compared to the checksum of the corresponding version in the **workspace stream**.
- In a *Modified* search, the checksum of each file element in the entire workspace tree must be compared to the checksum of the version in the workspace stream.
- The *Pending*, *Non-member*, *Overlap*, *Underlap*, and *Deep Overlap* searches also involve a search for modified files throughout the workspace tree.

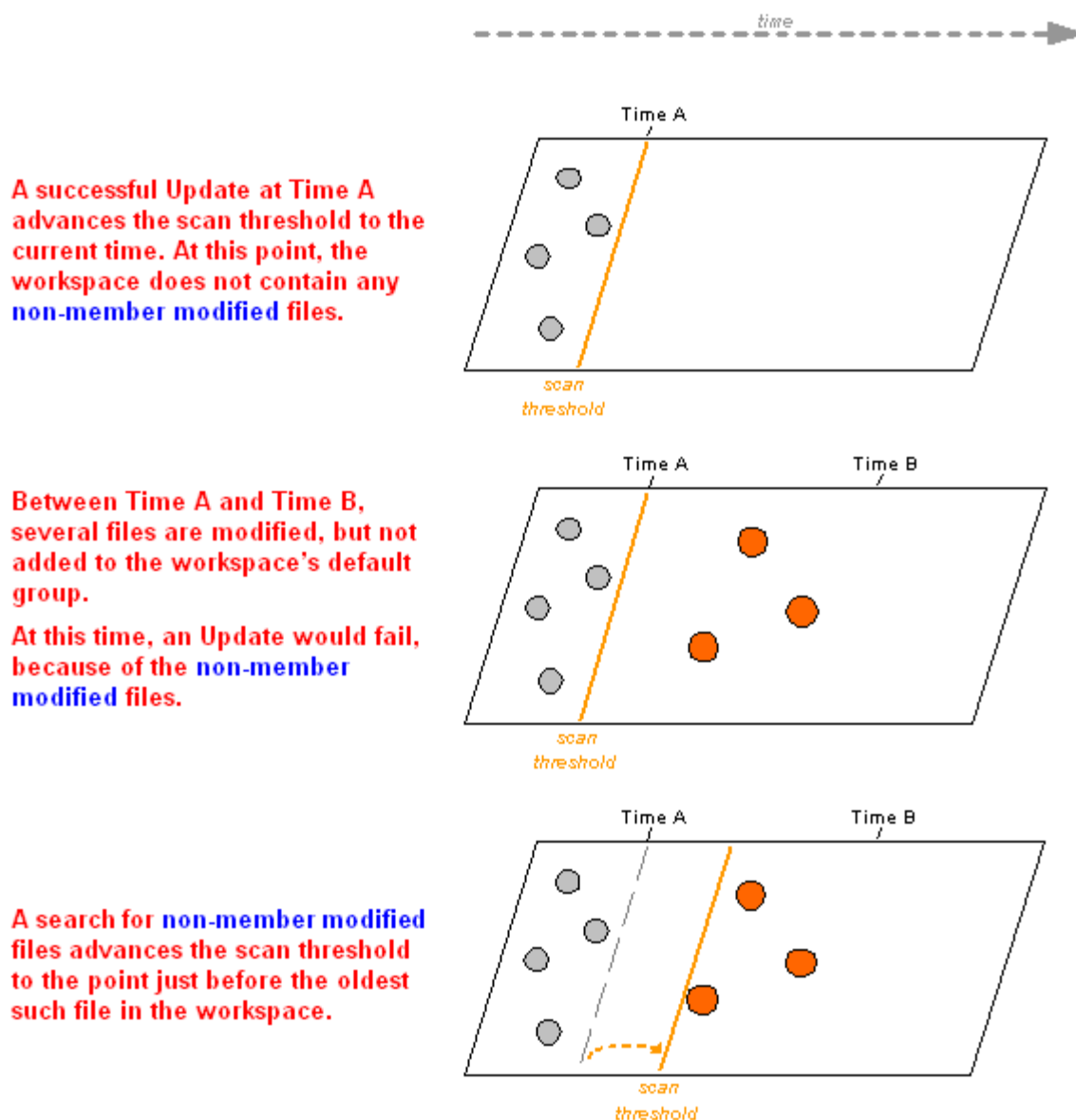
The *Update* command starts by performing a *Non-member* search, and refuses to proceed if it finds any modified files that are not in the workspace's default group. If the *Timestamp Optimization* checkbox (at the bottom on the File Browser tab) is checked, AccuRev skips the files in the workspace tree whose timestamps precede the workspace's **scan threshold**. This optimization can provide a significant performance increase.

### The Scan Threshold of a Workspace

AccuRev keeps track of such modified-file searches, maintaining an ever-advancing scan threshold for each workspace:

- After a successful update, the scan threshold is advanced to the time that the *Update* command began.
- After any of the searches listed above, the scan threshold is advanced beyond the time of the most recent update, to the point in time just before the timestamp on the oldest non-member modified file that the search located.

The scan threshold is not advanced when you work in the Folders pane, only the Searches pane.



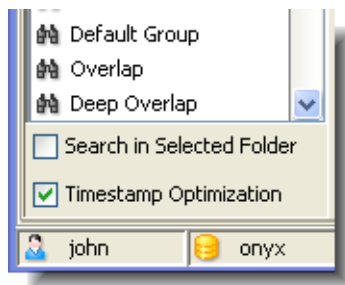
The intended effect of these manipulations is to set the scan threshold to the *latest* time for which this principle holds true:

For file elements that are not in the workspace's default group, the timestamp of a modified file is later than the workspace's scan threshold.

### Taking Advantage of the Scan Threshold: Timestamp Optimization

AccuRev's *timestamp optimization* feature takes advantage of the principle stated above. If the checkbox at the bottom of the File Browser tab is checked, AccuRev ignores files in the

workspace with timestamps preceding the scan threshold for all File Browser operations, including *Update*.



The advantage of using the scan threshold is simple: it enables AccuRev to ignore files and do its work more quickly. The later the scan threshold, the more files can be ignored, and the speedier the performance of the File Browser.

### Validity of Traditional Timestamp Optimization

The traditional *timestamp optimization principle* is usually valid -- but, unfortunately, not always. In general, if the only way you change version-controlled files is with text editors and build tools, the principle will be valid: each new change gets timestamped with the current time. But there are tools that can introduce "a new change with an old timestamp" into a workspace:

- The operating system's "copy file" command can preserve old timestamps when creating a new copy of a file. Similarly, the tar (UNIX) and zip (UNIX and Windows) utilities can preserve old timestamps when they copy files out of an archive.
- If environment variable ACCUREV\_USE\_MOD\_TIME is set, the Send to Workspace, Populate, Update, and the various Revert commands preserve timestamps when copying versions from the repository into a workspace.
- Less likely but possible, a severely-lagging system clock on an AccuRev client machine can cause edited files to get timestamps that precede the most recent update. (AccuRev commands won't execute if the client machine's clock is not synchronized with the server machine's clock. But something bad might happen to the client machine's clock at a time when no AccuRev commands are being executed.)

To work around these situations and continue using traditional TSO, clear the *Timestamp Optimization* checkbox before performing an *Update* or any of the relevant searches. This can slow performance significantly, but it guarantees that no modified file will be overlooked because of a misleading timestamp.

If you know exactly which modified files have old timestamps, you don't need to turn off the timestamp optimization. Instead, just update the timestamps to the current time, using the CLI command *accurev touch*.

### Local, Client-Based Timestamp Optimization

Because of the issues listed in the previous section, AccuRev 5.4 introduces a new, optional TSO methodology which makes use of cached server information that is stored on the client. This new approach to timestamp optimization ensures that all modified files in the workspace are correctly identified (even if they have an older time stamp and have been copied into the workspace from an external directory), while maintaining or increasing performance.

AccuRev CLI users access this new behavior through the stat command, or through commands such as anchor -n, co -n, files, and update (which make calls to the stat command). AccuRev Java GUI users access the behavior through the Update command and the File Browser Search Pane options (which make calls to the stat command).

### Local cache of element data

With local Timestamp Optimization, AccuRev maintains a cache database for each workspace which can be used for modified file detection, containing information about each file.

When TSO is enabled, this cache allows AccuRev to make a quick determination of modified state based on locally cached file sizes and timestamps without the hashcode computation and without going to the server.

The advantage of the new TSO approach is that time stamps for each individual file are compared to the cache, rather than comparing them to a single workspace scan threshold, which is set to the time stamp of the oldest modified file (as described in [The Scan Threshold of a Workspace](#) on page 98). This not only provides more valid results (by catching all modified files, even those with older time stamps that have been copied into the workspace from an external directory), it can be more efficient in situations such as a workspace that has a large number of (backed) files that have later time stamps than a single modified file.

The cache is updated by AccuRev commands that affect the file status or content on disk such as update, keep, add, pop, etc.

### Setting and Overriding Default TSO Behavior

The AccuRev administrator can set the default, site-wide timestamp optimization behavior by setting the following variable in the acserver.cnf configuration file:

- **STAT\_MODE=0** (or is not defined): Use traditional timestamp optimization.
- **STAT\_MODE=1**: No timestamp optimization.
- **STAT\_MODE=2**: Use the new, local timestamp optimization.

AccuRev CLI users can use or override the server STAT\_MODE settings by using the following switches with the stat command (or commands which call the stat command (anchor -n, co -n, files, and update):

- **stat** (no switch): use the default specified by STAT\_MODE
- **stat -O**: Do not use timestamp optimization (same as previous releases)
- **stat -O2**: Use the new, local TSO

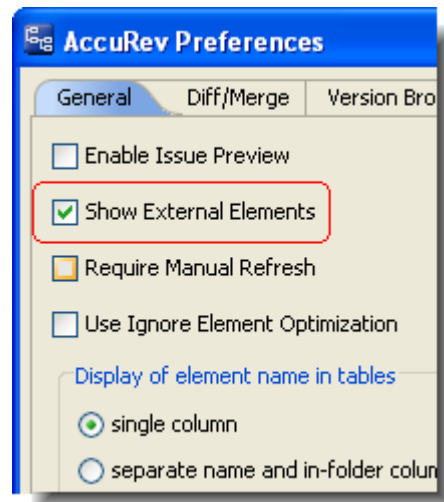
Note that if **STAT\_MODE=2**, there is no way for an AccuRev CLI user to override this and specify the traditional TSO.

GUI users cannot override the server STAT\_MODE setting, other than to not use any TSO, by turning off the Timestamp Optimization checkbox. This is the same behavior as in previous releases.

# File Browser: Controlling the Display of External Objects

Note: This topic applies to a File Browser displaying the contents of a **workspace** -- not a **dynamic stream** or **snapshot**.)

The Details pane of the File Browser can show both *elements* and *external objects*. The *Show External Elements* preference provides an all-or-none control over the display of external objects. Use the *Tools > Preferences* command to open the *AccuRev Preferences* window. See [AccuRev Preferences \(Tools > Preferences Command\)](#) on page 42



With a little more work, you can have the Details pane display some, but not all, external objects:

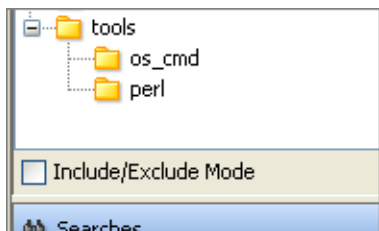
1. Set the *Show External Elements* preference.
2. Suppress the display of selected external objects through environment variable `ACCUREV_IGNORE_ELEMS`.

Also see [Updating a Workspace](#) on page 28.

## Using the File Browser's Include/Exclude Mode

By default, a workspace contains a copy of each file under version control in a particular depot. (More precisely, a workspace contains a copy of each version in a particular stream.) But sometimes, you don't want or need a copy of *every* file -- for example, if your current assignment doesn't include "rebuilding the world" each night. The depot might contain many thousands of files, of which you might need only a small subset. The File Browser's **Include/Exclude** facility enables you to define and work with a subset of the depot's elements.

When you check (or clear) the *Include/Exclude Mode* checkbox, the File Browser immediately switches into (or out of) Include/Exclude mode. (There's no *Ok* button to press.)

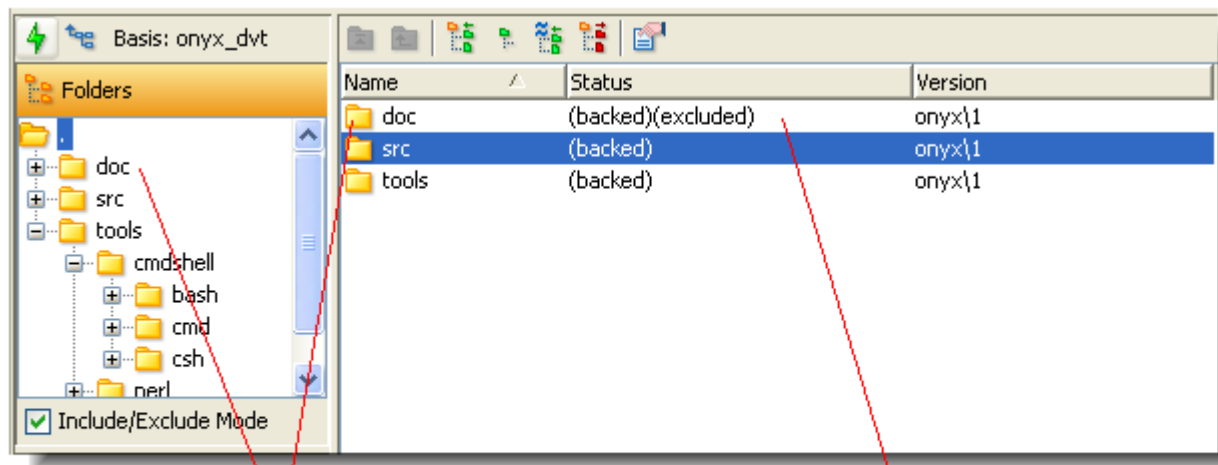


The File Browser works significantly differently in this mode, as described in the following sections.

See also: [Getting Only the Files You Need: the Include/Exclude Facility](#) on page 29 of the *AccuRev Concepts Manual*.

## File Browser Layout in Include/Exclude Mode

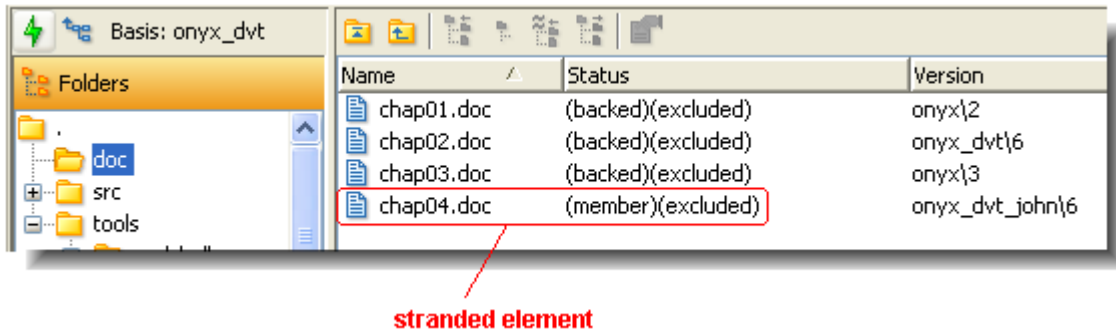
In Include/Exclude mode, both the Folders and Details panes show *all* elements in the workspace or stream. Elements that are currently excluded have **(backed)(excluded)** indicators in the Status column. For example, if top-level directory *doc* is excluded, the File Browser still shows it in both the Folders and Details panes:



excluded directory still appears in both  
Folders and Details panes

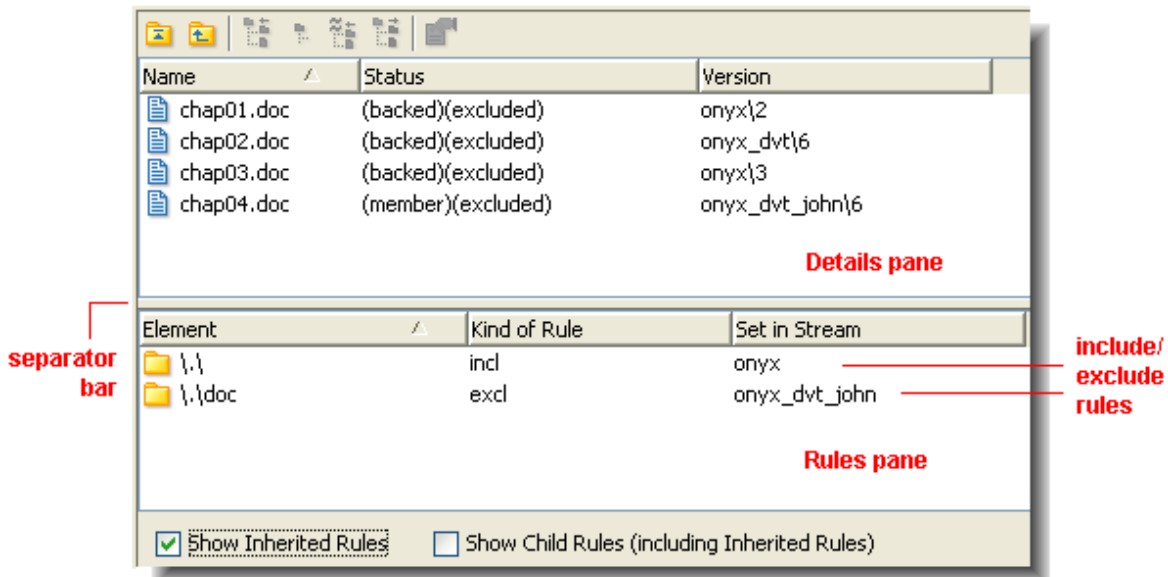
in Details pane, status is  
(backed)(excluded)

You can navigate into an excluded directory. Its elements have **(missing)(excluded)** status, except for those that are *stranded*, which have **(member)(excluded)** status.



The File Browser gets a new pane, the **rules pane**, in Include/Exclude mode. This pane shows include and exclude rules that affect the current workspace or stream:

- If the *Show Inherited Rules* checkbox is cleared, only the rules that were explicitly set for the current workspace are listed.
- If the *Show Inherited Rules* checkbox is checked, the listing also includes rules inherited from higher-level streams. The *Set in Stream* column indicates which higher-level stream.



In Include/Exclude mode, the Details pane toolbar changes: most of the buttons for AccuRev's version-control commands -- *Keep*, *Promote*, *Merge*, etc.-- disappear. Instead, buttons for the several *Include/Exclude commands* appear. A few of the standard toolbar buttons remain, to aid you in navigating the depot and determining element history.

## Working in the Rules Pane

The sections below describe how to work in Include/Exclude mode:

## Adding Rules

In each depot, there is one hard-coded *base rule*: the depot's base stream has an include rule that specifies the depot's top-level (root) directory. This rule makes the depot's entire directory hierarchy visible in the depot's base stream.

Element	Kind of Rule	Set in Stream
\\.	incl	eagle

Any number of rules can be added. Each rule applies to a particular pathname within the depot's directory hierarchy ("Element"), and applies at a particular level in the depot's stream hierarchy ("Set in Stream"). A rule set in a dynamic stream gets inherited by lower-level streams; but a rule for the *same element* in a lower-level stream or workspace overrides a rule in a higher-level stream.

Examples:

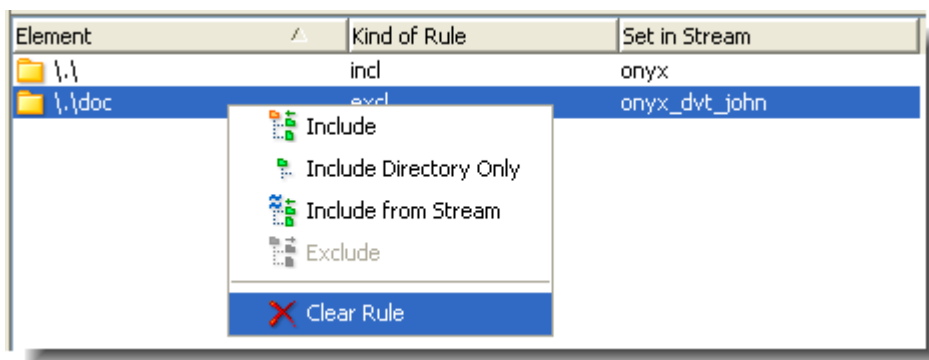
[Include/Exclude Example 1: Excluding a Directory](#) on page 108

[Include/Exclude Example 2: Simulating a Sparse Workspace](#) on page 109

## Removing Rules

To remove an existing rule that appears in the rules pane:

1. Make sure that your File Browser tab is displaying the stream or workspace in which the rule was explicitly set (*Set In Stream* column). If necessary, open a new File Browser on that stream or workspace.
2. Right-click the rule, and select *Clear Rule* from the context menu.



When you remove a rule from a stream, the effect is immediate on the stream itself and on streams below it. The effect does not take place on workspaces below the stream until they are *Update'd*.

When you remove a rule from a workspace, the effect is immediate on the workspace itself: files are copied into the workspace tree if you remove an exclude rule; files are deleted from the workspace tree if you remove an include rule.

## Leaving Include/Exclude Mode

To leave Include/Exclude mode in a workspace or stream, clear the checkbox at the bottom of the Folders pane. The rules pane disappears, and the Details pane reverts to:

- *not* displaying elements that have been excluded from the current workspace or stream
- including the standard Details pane toolbar

### Include/Exclude Mode Commands

The following commands are available in the File Browser's Details pane, when you have selected Include/Exclude mode:



#### Go to root

Have the Details pane display the contents of the depot's (and workspace's) top-level directory.

#### Up one level

Have the Details pane display the contents of the current directory's parent.

#### Include

Make the selected element(s) appear in the workspace or stream. If you specify a directory element, the entire subtree below it will appear in the workspace or stream, too.

As usual, the particular versions that will appear in the workspace or stream are the ones inherited from the *backing stream*.

#### Include Directory Only

Make the selected directory element(s) appear in the workspace or stream, but not any of the elements within the directory. You can use subsequent *Include* or *Include Directory Only* commands to make part of the directory's contents appear.

This command enables you to include files and directories that are deep in a depot's directory hierarchy, without having to include lots of other, unwanted elements. If you include a file or directory that is several levels below an include-only directory, AccuRev automatically creates the necessary *Include Directory Only* commands required to "dig down" to the level of the desired data.

Example:

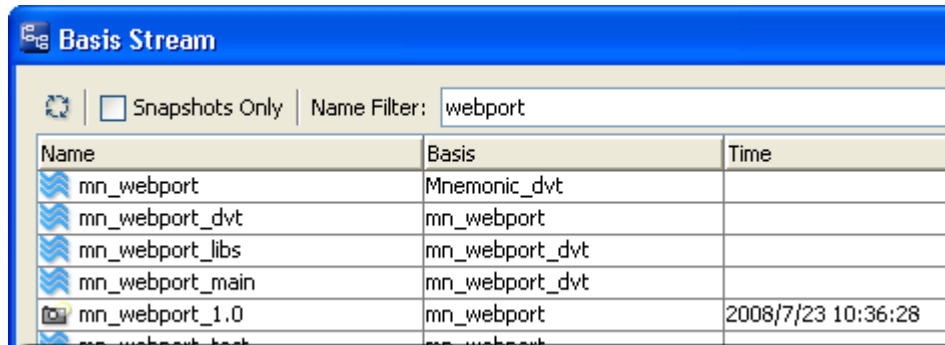
A depot contains a tools subdirectory, which contains a src subdirectory, which contain a scripts subdirectory, which contains two subdirectories, perl and python. If you need only the python subdirectory, not the rest of the tools subtree, use these commands:

- Invoke Include Directory Only on tools
- Invoke Include on python

In the Rules pane, note that AccuRev automatically creates "incldo" (Include Directory Only) rules for the intermediate directories, src and scripts.

#### Include from Stream

Establish a *cross-link*, making the selected element(s) appear in the workspace or stream. Instead of inheriting versions of the element(s) from the backing stream (as with the *Include* command), inherit the versions from the stream you specify in the Basis Stream dialog.



Use the toolbar to filter the streams shown in the dialog.

Note: For cross-linking to succeed, the element must have been present in the selected Basis Stream at the time your workspace was last updated. In a time-based stream, you cannot create a cross-link to another time-based stream (or to a snapshot) whose basis time is 'in the future' with respect to your stream.

- The *Refresh* button contacts the AccuRev Server to refresh the list of streams shown.
- The *Snapshots Only* checkbox restricts the streams shown to snapshot streams only. This setting (as well as the sort order of the columns in the dialog) is stored as a preference.
- The *Name Filter* field filters the streams shown using the text you enter in the field. If the stream name contains the text entered in this field, the stream appears. Otherwise, it is filtered out and does not appear.

If you specify a directory element, the entire subtree below it will appear in the workspace or stream, too.

Note: as of AccuRev 4.5: in a stream, you cannot *Promote* to cross-linked elements; in a workspace, you cannot *Keep* (or *Anchor*, or *Defunct*, etc.) cross-linked elements.

## Exclude

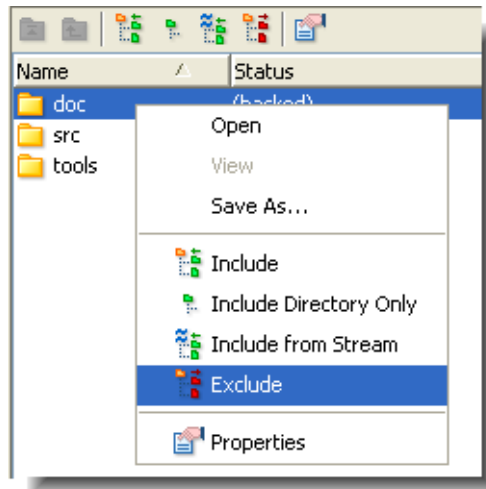
Remove the selected element(s) from the workspace or stream. For a workspace, the elements are deleted from the *workspace tree*. If you specify a directory element, the entire subtree below it will be removed.


## Properties

Displays information about the selected element: pathname, file type of current version, and element-ID.

## Include/Exclude Example 1: Excluding a Directory

To exclude a particular directory from the current workspace or stream, add a new rule for it:

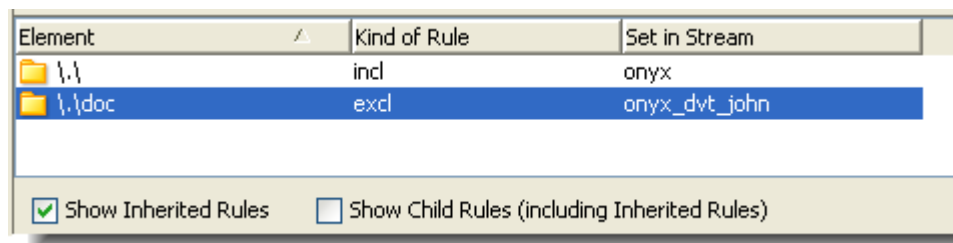


1. Make the directory appear in the Details pane, by going to its parent directory in the folders pane.
2. Select the directory to be excluded and click the  Exclude button in the toolbar. Alternatively, right-click the directory and select Exclude from the context menu.

Alternative: if a rule already exists for the directory -- set at the current level in the stream hierarchy or at a higher level -- you can invoke *Exclude* from the context menu of that rule in the rules pane.

When you exclude a directory, AccuRev removes the entire directory tree below it is removed from the workspace or stream. Popup windows have you confirm your intention and provide access to a log listing all the excluded elements.

The new exclude rule now appears in the rules pane:



If you're in a dynamic stream, AccuRev reminds you that elements won't be removed from workspaces below that stream until they are *Update* 'd. But keep in mind that element exclusion is instantly inherited by *streams* below that stream.

Note: If you want to remove some, but not all, of a directory tree from a workspace or stream, you must use an Include Directory Only rule, not an Exclude rule. See the other examples.

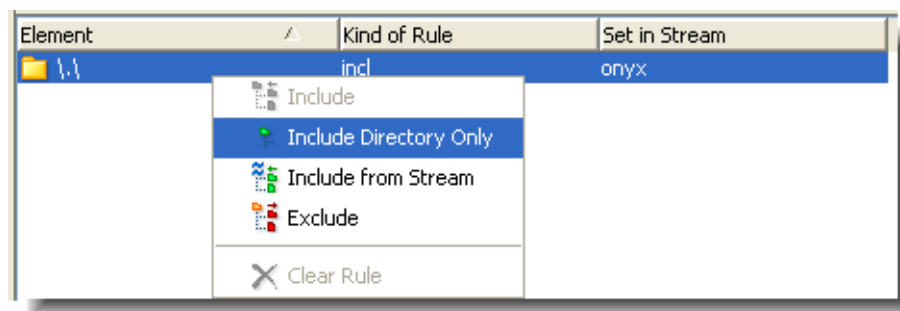
## Include/Exclude Example 2: Simulating a Sparse Workspace

If you're accustomed to working in a "sparse" workspace (supported in older versions of AccuRev), you may want to do the include/exclude equivalent in a new workspace:

- remove all elements from the workspace
- add back to the workspace just the elements that you want to appear in it

The first step is accomplished by setting an *Include Directory Only* rule for the depot's top-level directory. Since the top-level directory never appears in the File Browser's details pane, you must invoke the command from the rules pane, not the details pane:

1. Make sure that Show Inherited Rules is checked in the rules pane, causing the depot's base rule to be displayed.



2. Right-click the base rule, and select Include Directory Only.

This creates a new rule, at the workspace level, for the top-level directory. This rule overrides the rule you just right-clicked, and replaces it in the rules pane:



If you subsequently remove this new rule, the overridden rule will reappear in the rules pane, and will reapply to your workspace.

## The Add to Depot Command


The *Add to Depot* command places the selected files and/or directories under *version control*. For each file/directory, this includes:

- Creating an *element* in the depot.
- Creating the *version 1* of the element in your *workspace stream*.

The status of each file/directory changes from **(external)** to (kept).

### Invoking the Add to Depot Command

In the Details pane of a *File Browser* that is open on a workspace, select one or more files and/or directories. All the selected objects must currently have **(external)** status. Then:

- Click the  toolbar button, or
  - Right-click the selection and choose *Add to Depot* from the context menu.
- The *Add to Depot* dialog appears, which you use to complete the command.

## Using the Add to Depot Dialog

The *Add to Depot* dialog includes basic options (always visible) and advanced options (visibility controlled by a Basic/Advanced button).

### Basic Options

#### *Comment*

You can enter a comment string in the text box. (AccuRev "seeds" the text box with your most recent **Add to Depot** comment made during the current AccuRev GUI session.) The comment string becomes a permanent annotation to the version you're creating.

AccuRev itself does not require a comment, but your organization can establish such a requirement by putting a *trigger* on the *Add to Depot* command.

### Advanced Options

#### *Choose File Type*

By default, AccuRev determines the element type for the newly created version automatically (**text** or **binary**). You can override the default by specifying either of these types or **ptext** (a variant of **text**). See [AccuRev Element Types](#) on page 137.

#### *Choose Lock Type*

Choosing **Enable File Locking** places a *lock* on the element, to enforce *serial development* in *sibling workspaces* for that element. This ensures that users in sibling workspaces won't have to *merge* their work on this element. The lock remains on the element until you (or someone else) changes this value to **Disable File Locking** in a subsequent *Keep* command. See [The Locks Dialog Box](#) on page 53.

#### *Add links as symbolic links*

By default, a symbolic link (UNIX/Linux) or junction point (Windows) is added to the depot as an AccuRev element link, not as an AccuRev symbolic link. Use this checkbox to cause such objects to be converted to AccuRev symbolic links.

## Results of an Add to Depot Operation -- the Details

The *Add to Depot* command does the following:

- Creates a new element in the depot— A new **element** object is created in the **depot** to which your **workspace** belongs. AccuRev will track all future changes to this element -- both **content changes** and **namespace changes**.
- Creates the first version of the element—A new version object is created in the **workspace stream**. It has a unique **version-ID**, such as `talon_dvt_mary/1` ("the first version of this element created in workspace talon\_dvt\_mary"). A version object is immutable, and cannot be

removed from the repository. The version created by Add to Depot is termed a **real version**, because it represents an actual change to an element.

- Preserves the contents of the new version—The contents of the new version are preserved permanently in the repository. AccuRev copies the file in your **workspace tree** to a **container file** in the depot. No AccuRev operation modifies or removes the container file (exception: **archive** command).
- Sets the element status—The file element gets the status flag **(kept)** in your workspace. It also gets the **(member)** flag, indicating that the element is **active** in your workspace.

*Note: "Undoing" an Add to Depot operation*

There is no true "undo" operation in AccuRev version control, since that would violate the **TimeSafe** principle. But invoking the *Revert to Backed* command on a file that you just added to the depot (and have not promoted) is a very close approximation. The newly created element disappears from your workspace; you can retrieve its initial version only by using the unique **element-ID**. (CLI commands *hist -fv* and *cat -e*).

## The Anchor Command

The *Anchor* command makes the selected files **active** in the workspace (places them in the workspace's **default group**), without modifying them. Typically, you anchor a file in your workspace to prevent it from being overwritten with a newer version by a subsequent *Update* command. (*Update* overwrites inactive files only, not active ones.)


**Must-anchor situations:** You *must* anchor a file before editing it if

- You are working in an **anchor-required** or **exclusive file locking** workspace, or
- A development lock has been placed on that particular file element.

The *Send to Workspace* command is a variant of *Anchor*. Instead of activating the version that is currently in your workspace, *Send to Workspace* can activate any version of the element.

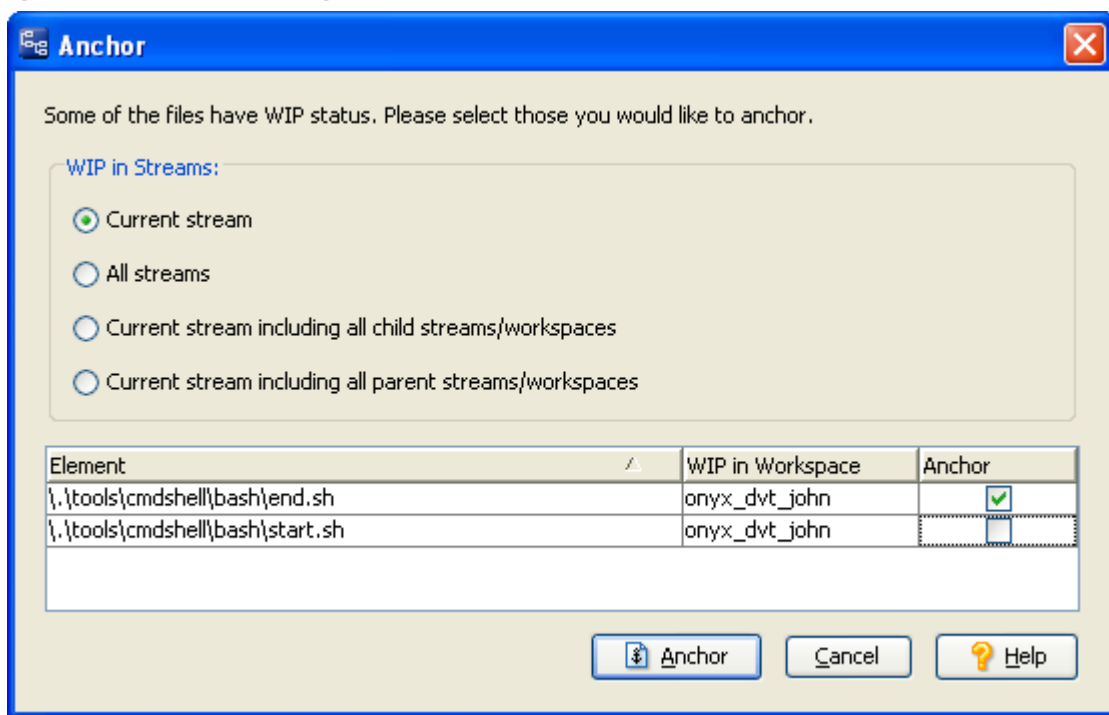
### Invoking the Anchor Command

In the Details pane of the File Browser, select one or more file elements whose current status is **(backed)**. Then, invoke the command in any of these ways:

- Click the  *Anchor* toolbar button.
- Right-click the selection to bring up its context menu, then choose *Anchor*.

In either of the must-anchor situations described above, if any element to be anchored is currently active in any **sibling** workspace, the *Anchor* dialog box appears, to help you complete the command.

## Using the Anchor Dialog



## The Annotate Tab


The *Annotate* command lists the entire contents of the selected version of a text file, annotating each line with:

- The timestamp of the version in which that line was added to the file.
- The number of the transaction in which that version was created.
- The user who performed that transaction.

A version timeline makes it easy to view the contents of *any* version of the text file.

### Opening an Annotate Tab

In the Details pane of the File Browser, select one text-file element. Then, invoke the command in any of these ways:

- Click the  *Annotate* toolbar button.
- Right-click the selection to bring up its context menu, then select *Annotate*.

### Annotate Tab Layout

The contents of a text file are listed as a multiple-column table. Each row represents one line of the text file.

Date	#	User	Line	Content
2004/11/08 23:21:53	33049	jpp	20	USER3 = "derek"
2005/03/14 15:47:19	34706	jpp	21	userlist = [USER1, USER2, USER3]
2005/03/14 15:47:19	34706	jpp	22	wkslist = {USER1:None, USER2:None, USER3:None}
2004/11/08 23:21:53	33049	jpp	23	
2005/05/24 17:02:40	37327	jpp	24	# global flag for renrandom()
2005/05/24 17:02:40	37327	jpp	25	# we want to perform renamings only in DVT stream hier,
2005/05/24 17:02:40	37327	jpp	26	RENAME_OK = False
2005/05/24 17:02:40	37327	jpp	27	
2004/11/08 23:21:53	33049	jpp	28	# data types
2004/11/17 16:22:18	33124	jpp	29	datatype = None
2004/11/08 23:21:53	33049	jpp	30	ZIP = 0
2004/11/08 23:21:53	33049	jpp	31	GEN = 1
2004/11/08 23:21:53	33049	jpp	32	SUFFIX = "txt"
2004/11/08 23:21:53	33049	jpp	33	SEP = os.sep
2004/11/08 23:21:53	33049	jpp	34	
2004/11/08 23:21:53	33049	jpp	35	# OS-specific settings
2004/11/08 23:21:53	33049	jpp	36	
2004/11/08 23:21:53	33049	jpp	37	if os.name == 'nt':
2004/11/08 23:21:53	33049	jpp	38	# where is AccuRev executable?
2005/05/24 17:02:40	37327	jpp	39	#os.environ['PATH'] = r'c:\ac37rc4\bin' + ';' + os.environ['PATH']
2005/05/24 17:02:40	37327	jpp	40	os.environ['PATH'] = r'c:\progra~1\accurev\bin' + ';' + os.environ['PATH']
2004/11/08 23:21:53	33049	jpp	41	# where will workspaces go?
2004/11/08 23:21:53	33049	jpp	42	rootdir = os.environ['SystemDrive']
2004/11/08 23:21:53	33049	jpp	43	# location of ZIP file containing soruces
2004/11/08 23:21:53	33049	jpp	44	BRASS_ZIPFILE = r"%s\AccuRev\brass.zip" % os.environ['m']

The columns contain:

- **Date** column: the timestamp of the *version* in which that line was added to the file.
- **#** column: the number of the *transaction* in which that version was created.
- **User** column: the AccuRev username of the *user* who created that version.
- **Line** column: the line number of the text
- **Content** column: the contents of the text line

In a **(modified)**-status file, one or more lines have not yet been saved with the *Keep* command. For such lines, columns 1-3 are empty.

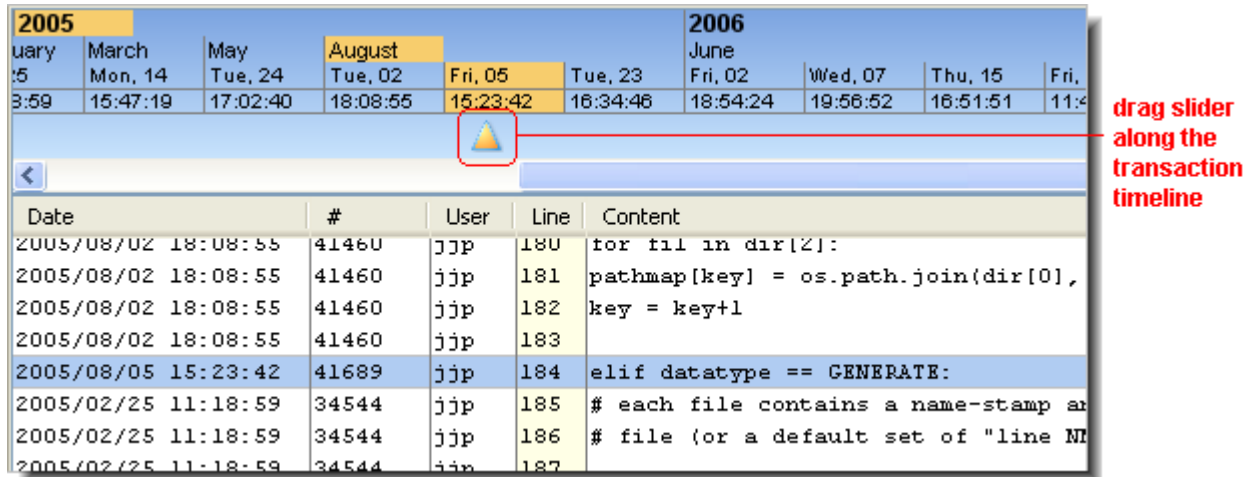
## Working in an Annotate Tab

When you click on a cell in the Date, #, or User column, AccuRev highlights all the lines whose value in that column matches the value in the cell. This makes it easy to:

- Find all the lines added or changed in a particular version (click in the Date or Transaction # column).
- Find all the lines created by a particular user (click in the User column).

## Using the Version Timeline

A timeline appears at the top of the Annotate tab, with an entry for each version of the text file element.



You can use the timeline's slider control to switch among the file's versions. (Alternatively, just click a timeline entry to jump to that version.) Whenever you switch to a different version, AccuRev automatically highlights the lines that were added or changed in that version.

## The Defunct Command

The *Defunct* command removes elements from active use in your workspace. That is, for each element you specify, it:


- removes the file, directory, or link from your *workspace tree*
- marks the element as **(defunct)** in the *workspace stream*
- makes the element *active* in the workspace, by adding it to the workspace's *default group*

Operating system commands won't find a defunct element (because it really is gone from the workspace tree), but the File Browser can see the element in your workspace stream, where it is both **(defunct)** and a **(member)**.

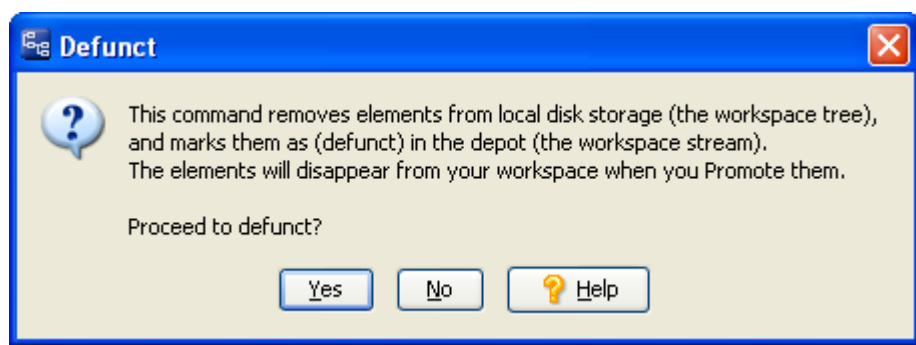
*Defunct* does not remove an element from the depot altogether (no operation removes an element altogether -- that would violate AccuRev's TimeSafe property). And it does not make an element disappear for all users. *Defunct* just removes an element from a particular workspace. The element remains visible in other streams and workspaces -- at least until you Promote it.

## Invoking the Defunct Command

In the Details pane of a File Browser that is open on a workspace, select one or more file elements. Then:

- Click the  *Defunct* toolbar button, or
- Right-click the selection and choose *Defunct* from the context menu.

The *Defunct* dialog appears, prompting you to confirm your intention to remove the element(s).



## Defuncting a Directory

If any of the elements you specify is a directory, *Defunct* works recursively: it removes the directory itself and elements under that directory. Only the specified directory itself becomes **(defunct)**; the objects below it are simply removed from the workspace tree, but do not become **(defunct)** in the workspace stream.

The precise result depends on whether any elements located below the defuncted directory are active:

- If no element below the defuncted directory is *active*, the entire directory tree is removed from your workspace tree.
- If one or more elements below the defuncted directory is active, those elements become *stranded*. A stranded element is *not* removed from your workspace tree, but its status becomes **(external)**, since the workspace stream no longer has a valid pathname to the object.

Caution: In all cases of defuncting a directory, a file below the defuncted directory that you have edited -- but never preserved with Keep -- will be removed from the workspace tree. (Such files are not officially "active" in the workspace.) This removes data for which there might be no other copy.

## Promoting a Defunct Element

After *Defunct*'ing an element, you use *Promote* to propagate its removal to your workspace's *backing stream*. The element won't appear in a File Browser that is open on your workspace, but it will appear, with **(defunct)** status, in a File Browser that is open on the backing stream.

## Bringing Back a Defunct Element

You can usually recover the backed version of the **(defunct)** element by invoking the *Revert to Backed* command on it. However, if either of the following conditions apply, the correct way to recover the version is to invoke the *accurev undefunct* command from the CLI.

- If you have kept changes in your workspace before defuncting. Remember that if you *Revert to Backed* an element which has not been *Promote*'d, you eliminate the version of that element in your workspace.
- If you have *Promote*'d the **(defunct)** element, so that it no longer appears in the File Browser.

## The Past and Future of a Defunct Element

When a user in a *sibling* workspaces performs an *Update*, the element will disappear from that workspace.

With each successive promotion up the *stream hierarchy*, a defunct element disappears from the source ("promote from") stream, and becomes *active*, with **(defunct)** status, in the destination ("promote to") stream.

AccuRev's TimeSafe property means that you cannot change the past. This means that a defunct element remains in old *snapshots* of streams. You can always get information about the element (if you know which stream it still exists in) using the *History Browser*.


## The Delete Command

The *Delete* command removes an object -- file, directory, or link -- from a *workspace tree*. It's equivalent to using an operating system command or tool (e.g. Windows Explorer, UNIX *rm* command) to delete the object. This command does not make any change in the AccuRev repository.

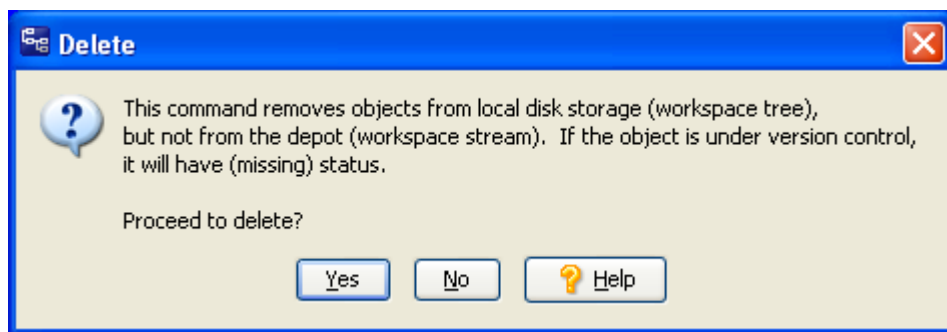
If the object is under *version control* (that is, it's an AccuRev *element*), its status becomes **(missing)**. If not, the object is simply removed from your workspace's local disk storage.

### Invoking the Delete Command

In the *Details* pane of a *File Browser* that is open on a workspace, select one or more objects. Then:

- Click the  *Delete* toolbar button, or
- Right-click the selection and choose *Delete* from the context menu.

The *Delete* dialog appears, prompting you to confirm your intention to remove the object(s).



## The Keep Command


The *Keep* command creates a new *version* of one or more file *elements*. Both the contents of the version and its identity are stored permanently in the AccuRev *repository*. For each file you *Keep*:

- AccuRev copies the contents of the file currently in your *workspace tree* to the associated depot in the repository.
- AccuRev assigns a *version-ID* to the new version, consisting of the workspace's name and an integer. For example, the version-ID

`talon_dvt_mary/17`  
... identifies the 17th version of the file element that was created in the workspace named *talon\_dvt\_mary*.

## Invoking the Keep Command

In the Details pane of a *File Browser* that is open on a workspace, select one or more file elements. Then:

- Click the  *Keep* toolbar button, or
- Right-click the selection and choose *Keep* from the context menu.

The *Keep* dialog appears, which you use to complete the command.

## The Keep Dialog

The *Keep* dialog includes basic options (always visible) and advanced options (visibility controlled by a Basic/Advanced button).

### Basic Options

#### *Comment*

You can enter a comment string in the text box. (AccuRev "seeds" the text box with your most recent **Keep** comment made during the current AccuRev GUI session.) The comment string becomes a permanent annotation to the version you're creating.

AccuRev itself does not require a comment, but your organization can establish such a requirement by putting a *trigger* on the *Keep* command.

### Advanced Options

#### *Choose File Type*

By default, AccuRev determines the element type for the newly created version automatically (**text** or **binary**). You can override the default by specifying either of these types or **ptext** (a variant of **text**). See [AccuRev Element Types](#) on page 137.

#### *Choose Lock Type*

Choosing **Enable File Locking** places a *lock* on the element, to enforce *serial development* in *sibling workspaces* for that element. This ensures that users in sibling workspaces won't have to *merge* their work on this element. The lock remains on the element until you (or someone else) changes this value to **Disable File Locking** in a subsequent *Keep* command. See [The Locks Dialog Box](#) on page 53.

## Results of a Keep Operation -- the Details

The *Keep* command does the following:

#### *Creates a new version*

A new version object is created in the *workspace stream*. It has a unique *version-ID*, such as `talon_dvt_mary/13` ("the 13th version of this element created in workspace `talon_dvt_mary`"). A version object is immutable, and cannot be removed from the

repository. The version created by *Keep* is termed a *real version*, because it represents an actual change to an element.

AccuRev allows you to invoke the *Keep* command on a file whose contents you haven't modified. So the new version might actually have the same contents as its predecessor.

### ***Preserves the contents of the new version***

The contents of the new version are preserved permanently in the repository. AccuRev copies the file in your *workspace tree* to a *container file* in the depot. No AccuRev operation modifies or removes the container file (exception: *archive* command).

### ***Sets the element status***

The file element gets the status flag **(kept)** in your workspace. It also gets the **(member)** flag, indicating that the element is *active* in your workspace. (It's possible that the element already had one or both these status flags, from previous activity in your workspace.) If the element previously had **(modified)** status, this flag is removed.

### ***Notes:***

#### ***Content changes and namespace changes***

The *Keep* command preserves *content changes* only, not *namespace changes*. Suppose you edit a file, so that it has **(modified)** status, then give it a new name with the *Rename* command. The new version that *Rename* creates does not record your content changes. The file will still have **(modified)** status until you preserve the contents changes with *Keep*.

#### ***"Undoing" a Keep operation***

There is no true "undo" operation in AccuRev version control, since that would violate the *TimeSafe* principle. The *Revert to Backed* command provides a close approximation. If you want to "undo" some, but not all of several intermediate *Keep*'s, perform a *Revert to Backed* command followed by a *Send to Workspace* of the intermediate version.


## **The New Folder Command**

The *New Folder* command creates a new, empty folder (directory) in the current directory ( the directory that's highlighted in the Folders pane ) of your workspace tree. It can also add the directory to the depot -- that is, place the directory under version control by converting it to an element. Note that **New Folder** is not available when you're working in the Searches pane.

If you choose not to place the directory under version control, you can do so later ...

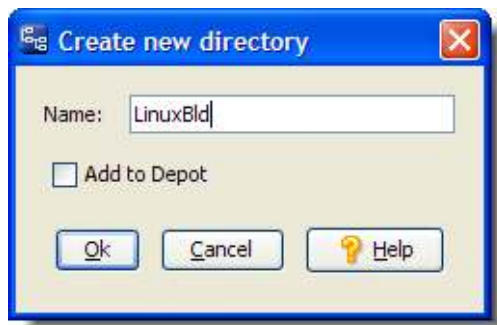
- explicitly, with the Add to Depot command
- implicitly, when you place any file or subdirectory below the new directory under version control

### **Invoking the New Folder Command**

In the Folders pane of the *File Browser*, navigate to the directory in which you want to create the new directory. Then, click the  *New Folder* button in the Details pane toolbar.

## Using the New Folder Dialog

You can specify these settings for the new directory:



### *Name*

The simple name (leaf name) of the new directory.

### *Add to Depot*


If checked, AccuRev will place the newly created directory under version control, by converting it to a directory element. The *Add to Depot* dialog appears, in which you can specify a comment. See [The Add to Depot Command](#) on page 109.

## The New File Command

The *New Folder* command creates a new, empty folder (directory) in the current directory ( the directory that's highlighted in the Folders pane ) of your workspace tree. It can also add the directory to the depot -- that is, place the directory under version control by converting it to an element. Note that **New File** is not available when you're working in the Searches pane.

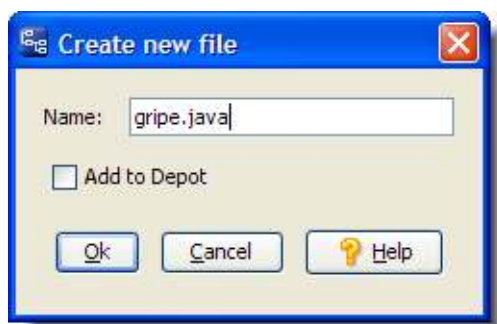
If you choose not to place the file under version control, you can do so later with the *Add to Depot* command.

### Invoking the New File Command

In the Folders pane of the *File Browser*, navigate to the directory in which you want to create the new file. Then, click the  *New File* button in the Details pane toolbar.

## Using the New File Dialog

You can specify these settings for the new file:



### *Name*

The simple name (leaf name) of the new file.

### *Add to Depot*

If checked, AccuRev will place the newly created file under version control, by converting it to a file element. The *Add to Depot* dialog appears, in which you can specify a comment, the file's *element type* and optional *exclusive file locking*.

## The Patch From Command

The *Patch From* (or *Patch*) command performs a **patch** operation on a text-file element, incorporating *some* of the changes in a specified "patch from" version into the version in your workspace. The combined contents are saved, with *Keep*, as a new version in your workspace.

The patch operation is operationally similar to a **merge** operation, which incorporates *all* of the changes in a specified version. (In fact, both operations use the same tool, either AccuRev's *Merge tool* or a third-party text-file-merge tool.)

### Patch vs. Merge

When you merge version V of a file into your workspace version, you are saying:

"combine my file with version V, taking into account all the changes in version V, back to the common ancestor"

When you patch version V of a file into your workspace version, you are saying:

"combine my file with version V, taking into account only the recent changes to version V"

To find the "recent changes to Version V", AccuRev scans backward through the file's *ancestry*, starting at version V and stopping when it encounters ...

- a version that was originally created in another workspace

... or ...

- a version that was promoted to another stream

This backward search defines a patch to the element. Version V is termed the **head version** of the patch. The older version is termed the **basis version** of the patch. The patch consists of all the versions of the element between the basis version and head version. The basis version itself is not included in the patch -- it precedes the set of "recent changes" in version V. The head version is included in the patch -- it contains the latest of the recent changes.

In identifying versions "between" the head and basis versions, AccuRev follows both direct ancestor and merge ancestry lines. In particular, if version V was created by a merge, its "recent changes" include both contributors to the merge.

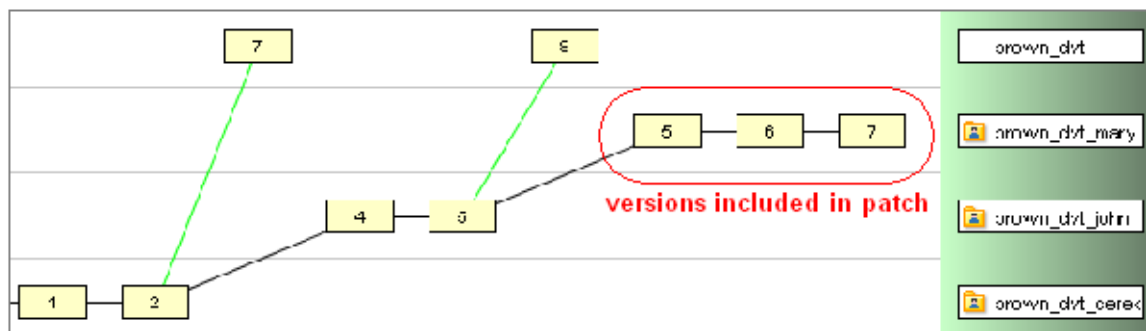
### Historical note

Prior to AccuRev 3.5.5, the Patch From command used a simpler definition of "recent changes to Version V" -- it meant "just the changes between version V and its immediate *predecessor*". (In effect, the *Show Patch List* command still uses this definition of patch, showing all the individual versions whose changes need to be propagated between two streams.)

For example, suppose user Mary recently created versions 4, 5, 6, and 7 of a file in her workspace, *talon\_dvt\_mary*. When patching version *talon\_dvt\_mary*/7 into your workspace version, AccuRev previously only incorporated the changes between versions *talon\_dvt\_mary*/6 and *talon\_dvt\_mary*/7. The limitation of this simple algorithm is clear: you probably wanted to incorporate all of Mary's recent changes — the modifications in versions 4, 5, 6, and 7. In previous releases, you would have had to perform four separate *Patch From* commands to achieve this result.

### Example 1

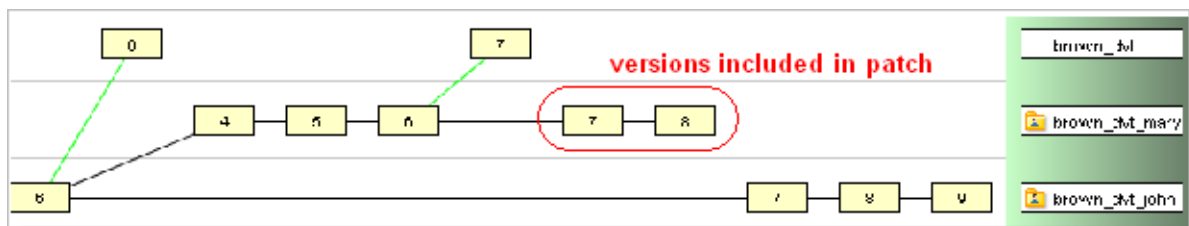
Before Mary started her recent work, she updated her workspace. This brought in a version of the file originally created by another user -- say, version *brown\_dvt\_john*/5. Then she proceeded to create versions 5 through 7 in workspace *brown\_dvt\_mary*.



Derek decides to patch version *brown\_dvt\_mary*/7 into his workspace. He invokes the *Patch From* command from the context menu of this version. AccuRev searches backward through the element's ancestry, and includes the set of changes recently made in Mary's workspace: this set includes *brown\_dvt\_mary*/5 through *brown\_dvt\_mary*/7; the patch doesn't include the two versions created in workspace *brown\_dvt\_john*.

### Example 2

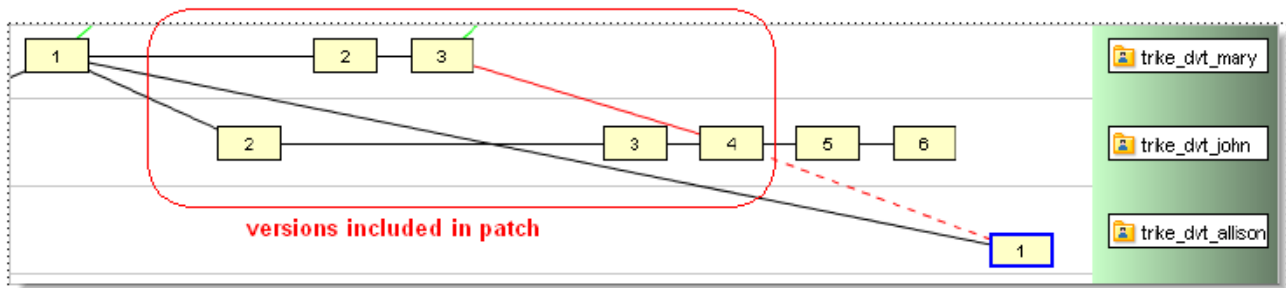
Mary started a task by bringing version *brown\_dvt\_john*/6 into her workspace with an update. Then she created versions *brown\_dvt\_mary*/4 through *brown\_dvt\_mary*/6, promoted her work to the backing stream, and then created two more versions: *brown\_dvt\_mary*/7 and *brown\_dvt\_mary*/8.



When John patches version *brown\_dvt\_mary*/8 into his workspace, AccuRev decides that only the versions since the promotion — versions 7 and 8 — contain "recent changes". The idea is that a promotion typically marks the end of a programming task, not an intermediate checkpoint.

### Example 3

John patches version *trike\_dvt\_mary*/3 into his workspace, creating version *trike\_mnt\_john*/1. Because Mary's version was created by a merge, the patch includes the recent changes to both merge contributors.



## Invoking the Patch From Command

You can invoke *Patch From* on a selected version in the following contexts:

- In the *Version Browser* (when invoked from a workspace), to patch from an arbitrary version.
- In the *History Browser* (when invoked from a workspace), to patch from the version in a particular transaction.

AccuRev processes the files you specify one-by-one. For each file, it does one or both of the following:

- Prompts you, in one or more dialogs, to resolve namespace-level conflicts.
- Invokes AccuRev's Merge tool or the other tool you've configured, in order to perform the patch operation. AccuRev's Merge tool opens in a separate tab within the AccuRev GUI window; any other tool opens in its own window.

## The Patch Algorithm

The *Patch From* command performs its content-level work with the Merge tool. Submitting a different set of versions to this tool effectively implements the patch algorithm (see [The Merge, Patch, and Reverse Patch Algorithms](#) on page 242).

## The Populate Command

For one or more selected elements, the *Populate* command copies the version that is currently in the *workspace stream* to the *workspace tree*. This has the effect of undoing an accidental deletion, or discarding edits that you have not yet preserved with *Keep*.

## Invoking the Populate Command

Select one or more file and/or directory elements in the Details pane of the File Browser. Then, choose *Populate* from the context menu of the selection.

## Using the Populate Dialog

You can select one or both of these options:

### *Recursive*

For each selected element that is a directory, perform a *Populate* on that element and on all elements below it.

### **Overwrite**

For each file element, replace the file (if any) currently in the workspace tree. By default, *Populate* silently declines to overwrite an existing file.

## **The Populate Progress Box**

The *Populate* command displays a progress box, much like the one used by the *Update* command.

## **The Promote Command**

The *Promote* command sends one or more *versions* from one *stream* to another, making the versions accessible in the destination stream -- and in workspaces and streams below it. A single transaction in the AccuRev depot records the creation of the new version(s) in the destination stream.

The basic use of *Promote* is to take the private versions that you've created in your *workspace stream* and make them public by sending them to the workspace's *backing stream*.

Promote can also send versions from a dynamic stream to any other dynamic stream. You can do such higher-level promotion using the *File Browser* or *StreamBrowser* tool. Promoting from a stream to a non-parent ("cross-promotion") requires use of an additional tool, the *Change Palette*.

## **Real versions and virtual versions**

The *Promote* command creates a *virtual version* in a *dynamic stream*. By contrast, the **Keep** command (as well as **Anchor**, **Send to Workspace**, **Revert**, **Rename**, **Defunct**, and the CLI-only command **undefunct**) creates a *real version* in a *workspace stream*. See [Real Versions and Virtual Versions](#) on page 137.

## **Inheritance of promoted versions**

After you promote a version, it can be inherited by all *dynamic streams* and *workspaces* below the *backing stream*. Inheritance by dynamic streams is automatic; inheritance by a workspace occurs when the user *Update's* it.

## **Overlap status and merging**

If an element has **(overlap)** status, you cannot promote it. You must *merge* your version with the version in the *backing stream*, to create a new, merged version. You can then promote the merged version.

## **Underlap status**


The status **(underlap)** is similar to **(overlap)**, in that an element has changed in your workspace and also in the backing stream. With **(underlap)**, the changes in your workspace version have already been promoted to the backing stream (from another workspace, or from a stream elsewhere in the depot's stream hierarchy). In many cases, it

is most appropriate to use the *Revert to Backed* command to 'undo' the changes in your workspace. In other cases, you can *merge* and *promote*, as with **(overlap)**-status elements.




## Invoking the Promote Command

You can promote versions in several contexts:


File Browser: Details pane



Select one or more elements in the Details pane of the File Browser. Then, click the  button in the Details pane toolbar, or choose *Promote* from the selection's context menu. This promotes the versions to the backing stream.

Stream Browser (graphical display only)

- Drag-and-drop the  icon below a workspace to the backing stream, to promote all the versions currently in that *default group*.
- Drag-and-drop the  icon below a dynamic stream to its parent stream to perform a standard promotion
- Drag-and-drop the  icon below a dynamic stream to any other dynamic stream to initiate a cross-promotion. The versions are loaded into the Change Palette, where you complete the promotion (perhaps after performing some *merges*).

Stream Browser: Default Group subwindow

In the graphical display, click the  icon below a workspace or stream to open a subwindow displaying the versions currently in the *default group* of that workspace or stream. Select one or more versions, then:

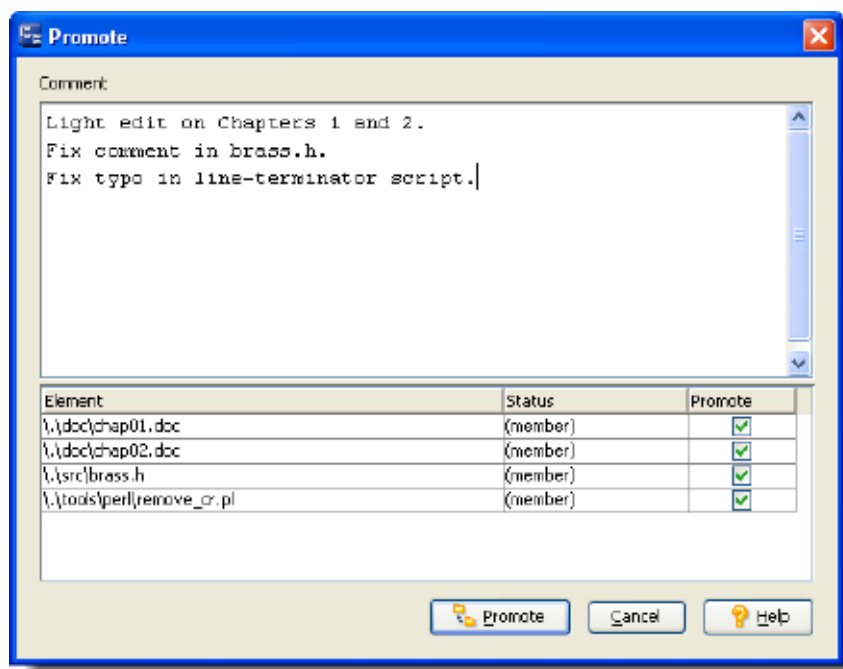
- Click the  button in the subwindow toolbar (or choose *Promote* from the selection's context menu) to promote the versions to the backing stream.
- Click the  button in the subwindow toolbar (or choose *Send to Change Palette* from the selection's context menu) to load the versions into the Change Palette where you complete the promotion (perhaps after performing some *merges*).

## Using the Promote Dialog

The *Promote* dialog provides these command options:

***Comment***

You can enter a comment string in the text box.

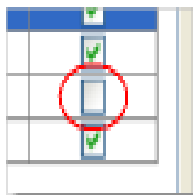


(AccuRev "seeds" the text box with your most recent **Promote** comment, made during the current AccuRev GUI session.) The comment string becomes a permanent annotation to the transaction that creates the new version(s) in the destination stream.

AccuRev itself does not require a comment, but your organization can establish such a requirement by putting a *trigger* on the *Keep* command.

### ***Promote (element list)***

(only for direct-promotions to the parent stream -- not for cross-promotions) All the versions you specified are listed in a table. You can deselect individual versions by clearing the checkbox in the table's Promote column. Deselected versions will not be promoted.



## **The Rename Dialog**

The *Rename* command changes the pathname at which an element is accessed. You can use this command to perform a simple renaming, to move an element to another directory, or both. You cannot move an element to another depot.

### **Invoking the Rename Command**

Select an element in the Details pane of the File Browser. Then, click the  button in the Details pane toolbar, or choose *Rename* from the selection's context menu.

## Using the Rename Dialog

Type a new pathname for the element:

- Type a simple name to rename the element, leaving it in the same directory.
- Type a relative pathname to move the element to another directory in the depot (and possibly rename the element, too).

(You cannot type a *depot-relative pathname*.)

In either case, a single **move transaction** records the changing of the element's pathname in your workspace.

## Renaming a Modified File Before Keeping It

You can create a new version of a file element in your workspace, recording either a *content change* or a *namespace change*. But you can't do both at once. For example, suppose you:

- 1.Edit the contents of file *color.java*, so that it gets **(modified)** status.
- 2.Use the Rename command to change the filename to *hue.java*.

This creates a new version of the file element in your workspace, recording the name change, but not the content change. To preserve the content change, you must use a separate *Keep* command to create a second new version.

## Caution on Reusing the Name of a Renamed Element

The ability to reuse the name of a renamed element provides significant flexibility for project refactoring tasks. But it also introduces a complication: what happens if you rename an element, create a new element at the same pathname, then invoke the *Revert to Backed* command on the renamed element?

The renamed element cannot revert to its old pathname, because there's a new element at that pathname. The original element simply disappears from your workspace. You might wonder "Why does the element not get (stranded) status?" The *Revert to Backed* command makes an element inactive in the workspace. The (stranded) status applies only to active elements.

At this point, your workspace contains a new element at the given pathname, and the parent stream contains the original element at that pathname. Attempting to promote the new element would produce a 'name already exists in parent stream' error. Use one of the following procedures to return this pathname to a consistent state:

### If you want to return to using the original element

- 1.Rename the new element, to a name like *myfile.java.DISCARDED*.
- 2.Defunct the new element

3. Promote the new element to the backing stream
4. At this point, the original element reappears in the Details pane, with **(missing)** status.
5. Invoke the Populate command on the original element.

### If you want to discard the original element and use the new element

These steps must be performed with the AccuRev CLI, which supports defuncting of the "disappeared" element using its *element-ID*.

1. Defunct the original element in the workspace, using the command **accurev defunct -e <element-ID>**.
2. Promote the defuncted version to the backing stream.

## The Revert Command

The *Revert* command implements an "undo change" capability for *dynamic streams*. It doesn't actually remove any versions from the stream or transactions from the repository; that never occurs, since it would be a violation of AccuRev's *TimeSafe* principle. Instead, *Revert* creates a new version of one or more elements in a workspace based on the stream.

NOTE: There is one exception to this workspace requirement: the Revert Change Package command option to continue without using a workspace, available in the StreamBrowser. (See [“Revert Change Package”](#) on page 161.)

The contents of the new version are the result of "subtracting out" a certain set of changes from the stream's current version:

- the changes that were added to the stream in a specified promote transaction, or
- the changes contained in a change package that is currently in the stream.

The "subtracting out" of content changes is performed by the Merge tool. Submitting a different set of versions to this tool effectively implements the *reverse patch algorithm* (see [Reverse Patch: Removing Content Changes](#) on page 246).

After the *Revert* command completes, you can verify the correctness of its work (and make further changes, if appropriate) in the workspace. Then, you can complete the "undo change to stream" process by promoting the new versions.

Note: Again, the [Revert Change Package](#) command available in the StreamBrowser has an option to continue without using a workspace. The results of this revert are placed directly in the stream itself. However, AccuRev recommends that you still build and test these changes in an updated workspace to verify the results.

### Invoking the Revert Command

There are several ways to invoke the *Revert* command:

• **Reverting a Transaction.** Open a *History Browser* tab, showing the transactions for the dynamic stream where you want to perform the "undo". To do this, choose **View > Streams** from the GUI main menu (and, if prompted, choose the depot containing your stream). Right click your stream, and choose **Show History** from the context menu. Then right-click the transaction to be reverted,

and choose **Revert** from the context menu. Only transactions whose action is **promote** can be reverted.

· **Reverting a Change Package.** Open a *Stream Issues* tab, showing the issues whose change packages are in the dynamic stream where you want to perform the "undo". To do this, choose **Show Issues** from the context menu of any dynamic stream in the StreamBrowser. Or choose **Show Diff by Issues** from the context menu of a dynamic stream, then select another stream to compare it with.

## Choosing a Workspace

AccuRev displays the workspaces that both:

- ... belong to you, and
- ... are children of the stream whose transaction or change package you wish to revert

The workspace you select will be used to perform a *reverse patch* operation. AccuRev invokes the Merge tool in a special mode to perform this operation, with the file in the selected workspace as one of the contributors. It refuses to proceed if the file is currently *active* in the workspace.

## Choosing a Workspace for the Revert Command

AccuRev displays the workspaces that both:

- Belong to you, and
- Are children of the stream whose transaction or change package you wish to revert

The workspace you select will be used to perform a *reverse patch* operation. AccuRev invokes the Merge tool in a special mode to perform this operation, with the file in the selected workspace as one of the contributors. It refuses to proceed if the file is currently *active* in the workspace.

## Revert to Backed Dialog

For each selected element, this command remove all changes you've made since the last time you Promoted it, or since the last Update of the workspace -- whichever is more recent:

- File elements: this includes both content changes and namespace changes.
- Directory elements: this includes namespace changes only.
- Link elements: this command operates on the link itself, "undo"ing changes you've made to the specification of the link target.

The selected elements become inactive in the workspace (they are removed from the workspace's default group). The status of the elements becomes **(backed)**.

For a detailed discussion of the Revert to Backed command, see [Chapter 7 Notes on Revert to ... and Diff Against... GUI Commands](#) in the AccuRev [Technical Notes](#).

## Trigger Firing

When you invoke this command in a dynamic stream, the depot's *pre-promote-trig* trigger fires. That's because, like *Promote*, this command changes which version of the element appears in the stream.

## Revert to Most Recent Version Dialog

(This command is enabled only in a workspace, and only for file elements with **(modified)** status. Link and directory elements cannot have **(modified)** status.)

This command replaces (by invoking the [Populate](#) command) each selected file with the most recent version you created with *Keep*. Use this command when you've saved one or more intermediate versions of the file(s), and you want to discard further changes you've made since a *Keep*.

The selected elements remain active in the workspace. That is, they are *not* removed from the workspace's default group. The **(modified)** status of the elements changes to **(kept)**.

If you've modified a file but not yet performed a *Keep* on it, this command works like the [Revert to Backed Dialog](#).

## The Send to Issue Command

The *Send to Issue* command creates or modifies *change package* entries for one or more elements in one or more issue records. When first created, a change package entry has the same structure as a *patch*: the set of 'recent changes' to an element. After subsequent modifications, a change package entry can include a much broader set of changes.

The command has several variants, providing different ways to specify the data to be sent to the change package(s):

- You specify a version, which AccuRev interprets as the **head version** of a patch. AccuRev automatically determines the corresponding **basis version**.
- You specify both the head version and basis version of a patch.
- You specify an existing change package entry, to be sent to one or more additional change package(s).


### Invoking the Send to Issue Command

You can invoke *Send to Issue* in the following contexts. Note that each context provides access to some, but not all, variants of the command.

- In the Version Browser, choose Send to Issue or Send to Issue (specifying basis) from the context menu of any version.
- In the Versions pane of the History Browser, select one or more versions and choose Send to Issue from the selection's context menu.
- In the Stream Version Browser, choose Send to Issue from the context menu of any version.
- In the Changes subtab of an issue record or the Change Package Contents pane of a Stream Issues tab, select one or more change package entries and choose Send to Issue from the selection's context menu.

### Specifying the Basis Version of the Change

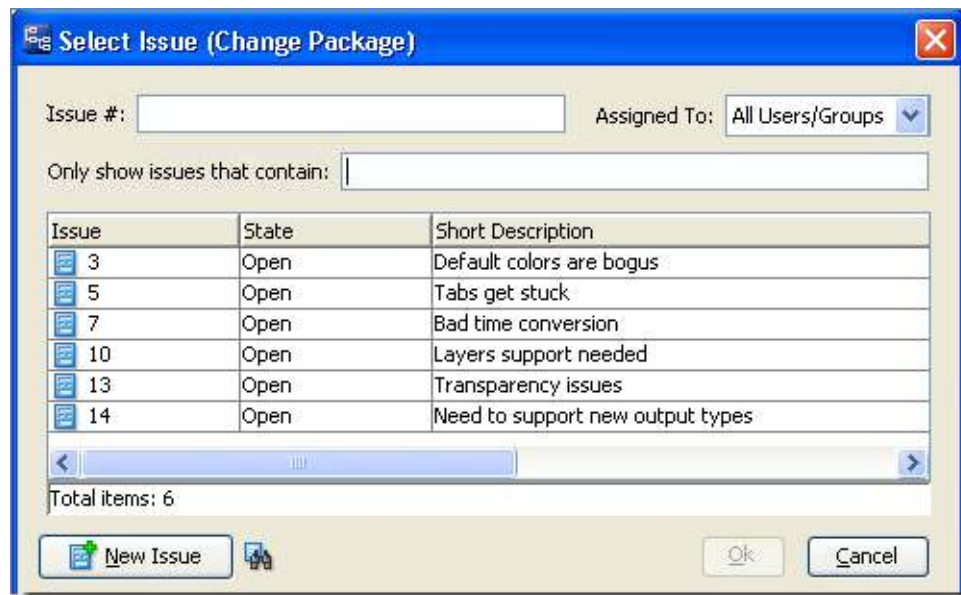
In the *Send to Issue (specifying basis)* variant of the command, you specify the basis version, defining the extent of the change to be sent to the change package(s). After you invoke the

command, the mouse pointer changes to . Click the version you wish to be the basis version for the change. (You can press the **Esc** key at this point to cancel the command.)

The basis version you specify must be an *ancestor* of the head version. If it isn't, an "Invalid Change Package" error occurs.

## Using the Send to Issue Dialog

AccuRev prompts you to specify one or more issue records, in which change package entry(s) will be created or modified.



If you've specified one of your **AccuWork queries** as the **default query**, AccuRev executes the query and displays the results.

You can select one or more of the issue records:

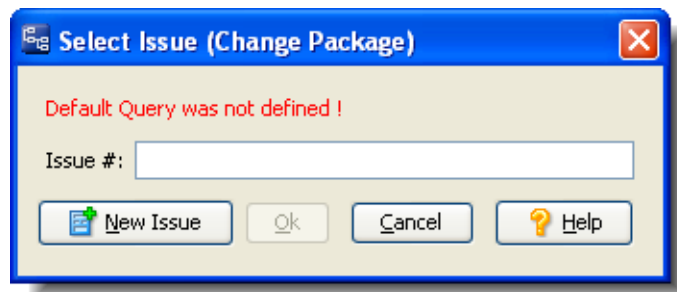
- By clicking: left-click to select an issue record. Hold down the Ctrl or Shift key to select multiple records.
- By typing: enter one or more numbers, separated by SPACES.
- By filtering: enter text into the "Only show issues that contain" field, then click to choose an issue from the filtered list. Delete the text in that field to remove the filter and display all issues.

You also have two other options for specifying issue records other than the initial set shown by the dialog::

- Choose an issue record associated with another user. To do so, choose a user or group name from the drop-down list to the right of the Issue # field. The set of names shown here is defined when the change-package configuration is set (see [The AccuWork Schema Editor \(Change Packages subtab\)](#) on page 325).
- Create a new issue using the New Issue button at the bottom of the dialog. Note that if you have chosen to display issue records associated with another user, the new issue must be

associated with that user when you create it, or the issue will not display on the Send to Issue dialog. Note: If you have AccuWorkflow enabled, and use a New Issue button from within a dialog in the Java client, you must set any AccuWorkflow fields in those issues manually.

If you don't have a default query, AccuRev prompts you to type in an issue number (or multiple numbers, separated by SPACES).

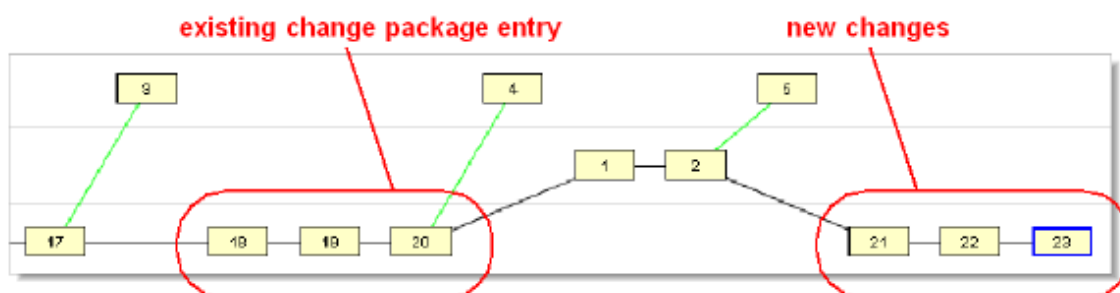


## Failure to Modify an Existing Change Package Entry

The *Send to Issue* command fails if AccuRev determines that an existing change package entry cannot be modified by "adding in" the set of versions you specified. A change package entry must always be expressible as "all the versions between the specified head and basis versions". Here are two situations in which this is not possible:

### Gap in change package entry

This case typically arises when you do work on a task at two different times, and someone else has worked on the same element in between. In this example, you created versions 18-20 of a file and created a change package entry with those versions. A colleague updated her workspace to bring in your changes, created versions 1 and 2 in her workspace, and promoted her changes. You then updated your workspace to bring in her changes, and made a new set of changes, in versions 21-23.



Now, you want to add your new changes into the same change package. When AccuRev tries to combine the six versions into a single change package entry, it detects the 'gap' consisting of your colleague's versions 1 and 2, and displays an error box.

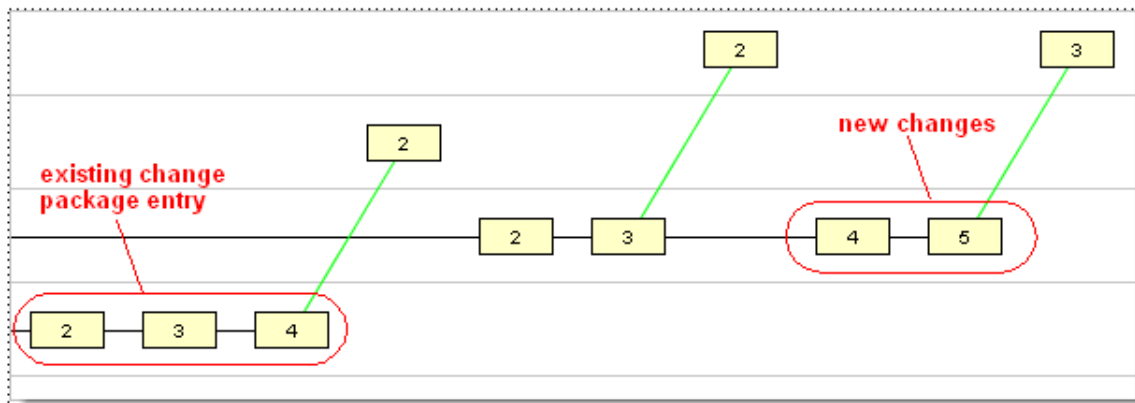
### Fixing the Problem

Typically, you'll want to "span the gap", making the change package entry include the entire series of versions — your original changes, your colleague's changes, and your new changes. You can accomplish this by first removing the existing change package entry from the *Changes subtab* of the issue record, then using the *Send to Issue (specifying basis)* variant of the command. In this example, you'd specify version 23 as the head version and version 17 as the basis version.

Note: You can accomplish the same result even more easily if you encounter this situation while executing the Promote command on version 23, and you are using the change-package-based integration between AccuRev and AccuWork. In this case, AccuRev offers to 'span the gap' between the two change set entries automatically.

### Merge required before change package entry can be modified

This case typically arises when developers in different workspaces try to use the same change package. There is no simple "gap" between the existing change package entry and the new set of version.



This means there is no way to combine them into a single change package entry.

### Fixing the Problem

You can remedy this situation by performing a merge at the element level. (There is no merge operation defined at the change package level.)

## The Send to Workspace Command

The *Send to Workspace* command *activates* the specified version of an element (or set of elements) in a particular workspace. All of the elements must currently be inactive that is, have **(backed)** status -- in the workspace. This command:

- Copies the contents of the specified version from the **repository** to the **workspace tree**.
- Changes the status of elements to **(member)**, indicating that it is in the workspace's **default group**.

This command is a more generalized variant of the *Anchor* command, which always operates on the version of an element that is currently in the workspace.

### Invoking the Send to Workspace Command

You can invoke *Send to Workspace* in any of these contexts:

- In the Version Browser, choose Send to Workspace from the context menu of any version.
- In the Versions pane of the History Browser (displaying the history of an individual element), select one or more versions and choose Send to Workspace from the selection's context menu.
- In the Stream Version Browser, choose Send to Workspace from the context menu of any version.

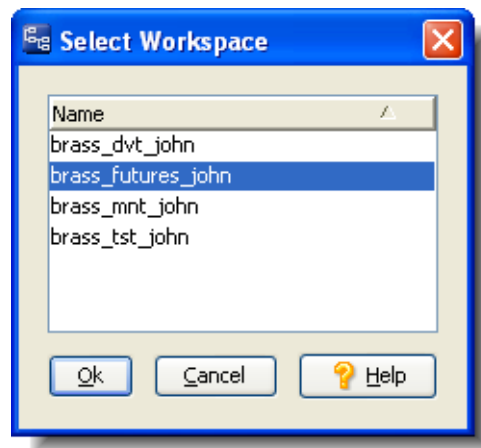
In each of the above cases, you have a current workspace context -- the workspace name is displayed at the bottom of the GUI window. AccuRev sends the selected versions to this workspace.



You can also invoke this command from the *Summary pane* of the History Browser. Select a single transaction and choose *Send to Workspace*. In this case, you don't necessarily have a current workspace context, AccuRev prompts you to select one of the workspaces belonging to you. All the versions in the selected transaction are sent to the workspace you select.

## Using the Send to Workspace Dialog

Specify a workspace by:



- double-clicking it, or
- selecting it and clicking Ok

You can click the **Name** column header to reverse the sort order of the workspace names.

## The Update Progress Box

The *Update* and *Update Preview* commands open a pop-up window to display the command's progress. This window displays the filenames being processed one-at-a-time or in a scrolling text pane, depending on the setting of the *Show Progress Log* user preference.

You can use the following command buttons in the progress box:

### View Full Log

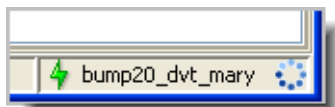
(enabled after command completes) Display the entire *Update* or *Update Preview* log in a separate text-editor window.

### Save Log As

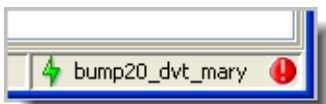
(enabled after command completes) Create a text file containing the entire *Update* or *Update Preview* log.

## Run in Background

Close the progress box, but continue command execution. You can proceed to perform other work in the AccuRev GUI window. A progress indicator appears in the lower right corner of the GUI window.



The indicator disappears when the command completes. If an error occurs during command execution, the progress indicator changes to a **flashing red** error indicator:



Click on the indicator to restore the progress box. The update log will include a description of the error.

### Cancel (while command is in progress)

### Close (after command completes)

Close the progress box. If the *Update* or *Update Preview* command is still executing, cancel it.

## The WIP (Work in Progress) Tab

The WIP ("Work in Progress") tab shows elements that are currently *active* in the workspaces below a specified set of one or more streams. (A stream's own *default group* is not included in this listing.)

### Opening a WIP Tab

Select a dynamic stream in the StreamBrowser, and choose *WIP* from its context menu.

### WIP Tab Layout

WIP in Streams:		
<input type="radio"/> Current stream <input type="radio"/> All streams <input checked="" type="radio"/> Current stream including all child streams/workspaces <input type="radio"/> Current		
Element	WIP in Workspace	
\\.\dir01\sub00\file47.txt	jumbo_dvt_dan	
\\.\dir01\sub02\file35.txt	jumbo_dvt_dan	
\\.\dir04\sub03\file17.txt	jumbo_dvt_dan	
\\.\dir06\sub00\file09.txt	jumbo_dvt_dan	

#### *Element (or separate "Name" and "In Folder" columns)*

The element's pathname within the depot.

#### *WIP in Workspace*

The workspace in which the element is active.

## Working in a WIP Tab

Use radio buttons to specify which stream -- or set of streams -- whose workspaces are to be searched for active elements:

### *Current stream*

The stream you selected when invoking the *WIP* command.

### *All streams*

All streams in the depot.

### *Current stream including all child streams/workspaces*

The entire subhierarchy of streams below the one you selected.

### *Current stream including all parent streams/workspaces*

The entire *backing chain* of the stream you selected.

A workspace is searched for active elements if it is a direct child of one of the streams you've indicated with your radio-button choice. (But *pass-through streams* are ignored when determining the "direct child" relationship. Workspaces whose parent is a pass-through stream are treated as if they were backed but their "grandparent" stream.)

The WIP tab display is informational only. You cannot change the status of elements from this tab.

See also: [Working with Tables](#) on page 9

## The Default Group of a Workspace or Stream

For each *dynamic stream* and *workspace* (but not for *pass-through streams*, *snapshots*, or *reference trees*), AccuRev keeps track of the elements that are currently under *active* development in that context. This set of elements is called the **default group** of the stream or workspace.

Note: For some purposes, the default group is regarded as containing particular versions of the elements -- for example, when you promote a stream's entire default group.

## How Elements Become Members of the Default Group

### *Workspace*

The default group is a property of the *workspace stream*, maintained in the AccuRev repository. There are no "hidden files" or other administrative structures in the *workspace tree* to keep track of which elements are active.

The following commands cause an element to become a member of the workspace's default group. Note that these commands can be executed *only* in a workspace, not in a stream.

- Keep
- Rename (or cut-and-paste)
- Defunct
- undefunct (CLI only)

- Merge (ends by performing a Keep)
- Revert (ends by performing a Keep)
- Paste Link (CLI command: ln)
- Send to Workspace / Anchor

### ***Dynamic Stream***

When a version is *Promote*'d to a dynamic stream, the element becomes a member of the stream's default group.

If an element is already in the default group of a workspace or stream when one of the commands listed above is executed, the element simply remains a member. In other words, the default group's membership doesn't change when it's viewed as a set of elements. But it does change if you view the default group as a set of particular versions of those elements.]

### **How Elements are Removed from the Default Group**

For both a workspace and a dynamic stream, the following commands remove an element from the default group:

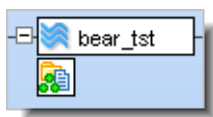
- Promote (to the immediate parent stream)
- Revert to Backed

The element returns to a *passive* state in the workspace or stream. A stream returns to inheriting a version of the element from its parent stream; a workspace returns to acquiring versions of the element from the backing stream through the *Update* command.

Cross-promoting an element to a stream that is *not* the immediate parent stream does not remove the element from the default group of the "from" stream.

### **Viewing the Contents of the Default Group**

The *Stream Browser* can show the *development activity* currently taking place in each stream or workspace.



A control below the stream or workspace opens or closes a subwindow that displays the details of the development activity. The activity details can appear in several ways -- by element, by transaction, or by issue record.

For a dynamic stream, the by-element display is a listing of the stream's default group. For a workspace, this subwindow displays those members of the default group that would be found in a *Pending* search (see discussion below).

The *File Browser's Default Group* search displays *all* the members of the default group of a stream or workspace.

### **Active Versions vs. Pending Changes in a Workspace**

A workspace's set of active elements (that is, its default group) is not quite the same as the set of elements with pending (not yet promoted) changes. In practice, the sets will often be the same. But in theory, the sets are overlapping, but not identical:

- The Default Group search can include elements that are not pending. These are elements that you've "checked out" with Send to Workspace or Anchor.
- The Pending search can include elements that are not in the default group. This is the set of a "non-member" elements -- those that have been modified (and thus are pending), but are not in the default group.

## AccuRev Element Types

Each *version* of a file *element* has an **element type**, which identifies the type of data contained in that version. Each time you create a new version (with the initial *Add to Depot* and subsequent *Keep*'s), you can change the element type, using the *Advanced* section of the command's dialog box.

AccuRev supports the following element types. For a newly created version of any type, AccuRev creates a new *storage file* in the *file storage area* of the AccuRev *repository*.

### *binary*

When a new **binary** version is created, the storage file is an exact copy of the file in the *workspace tree*. When you retrieve a binary version from the repository (for example, with *Send to Workspace* or *Update*), AccuRev makes an exact copy of the storage file in the workspace tree.

### *text*

When a new **text** version is created, the storage file is created with a single **NL** character (hex character code **x0A**) at the end of each text line. This means that a version's storage file may have different line terminators than the workspace-tree file you submitted to the command.

When you retrieve a text version from the repository (for example, with *Send to Workspace* or *Update*), the file created in the workspace tree uses either the UNIX/Linux line terminator (**NL** / hex **x0A**) or the Windows line terminator (**CR-LF** / hex **x0D-x0A**) - depending on the EOL setting for your workspace (see [The New Workspace Command: Screen 3 of 3—Configuring the Workspace](#) on page 22).

### *ptext*

**ptext** versions are handled like **binary** versions: exact copies are transferred between the workspace tree and the file storage area. The difference between **ptext** and **binary** files is in how they are handled by the *Merge* and *Patch* commands.

## Real Versions and Virtual Versions

The difference between the *Keep* and *Promote* commands highlights an important aspect of the way that AccuRev organizes and manages development data. It also highlights the difference between *backing streams* and *workspace streams*.

## Real Versions

All 'real' development -- both *content changes* and *namespace changes* -- takes place in workspaces, not in *dynamic streams* or *reference trees*. The following commands create a new **real version** of some element in the built-in *workspace stream* of some AccuRev *workspace*.

- Keep*
- Rename* (or cut-and-paste)
- Defunct*
- undefunct** (CLI only)
- Merge* (ends by performing a *Keep*)
- Revert* (ends by performing a *Keep*)
- Paste Link* (CLI command: **ln**)

## Virtual Versions

Unlike the commands listed above, the *Promote* command does not record a change to the contents or pathname of any element. Rather, it makes an existing real version available in a new location: a particular *dynamic stream*. The version that *Promote* creates in a dynamic stream is called a **virtual version**; a new virtual version is a pointer to, or alias for, an existing real version in some workspace stream.

### Virtual Versions in Workspace Streams

The commands *Send to Workspace* and *Anchor* are special. They create a new version in a workspace stream, but the new version is *virtual*, not real. Why? It's for consistency with the above descriptions of real and virtual versions:

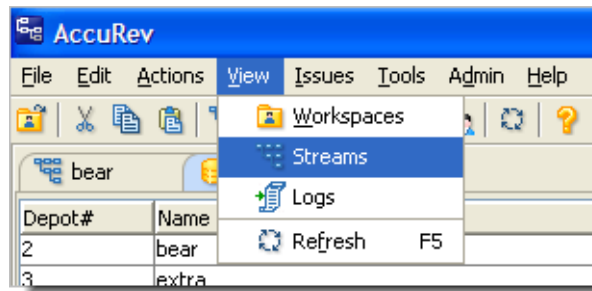
- These commands do not preserve a development change, so it doesn't make sense to create a new real version.
- These commands make an existing real version available in a new location: a particular workspace -- so it makes sense to record the action as a virtual version.


## 4. The StreamBrowser

AccuRev's unique StreamBrowser enables graphical control over the entire configuration management environment, in a way that is simple, flexible, and powerful. Using the StreamBrowser, you can view and manipulate all the streams in a depot, including *workspace streams* and *snapshots*.

### Opening a StreamBrowser Tab

You can open a StreamBrowser tab to display the streams of the *current depot* in any of these ways:

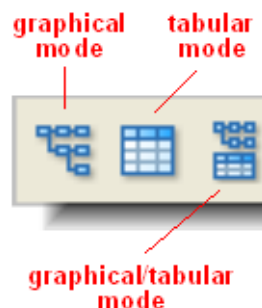


- Choose *View > Streams* from the GUI main menu.
- Click the  *View Streams* button on the GUI main toolbar.
- Choose *View > Streams* from the context menu of a depot on a *Depots* tab.

If there is no current depot, AccuRev prompts you to select one.

### StreamBrowser Tab Layout

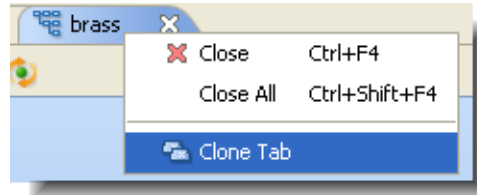
You can use the StreamBrowser in any of the following display modes, which you select using buttons on the *StreamBrowser toolbar*:



- *Graphical mode*: displays the depot's stream hierarchy as a tree (root node at the upper left)
- *Tabular mode*: displays a table containing the depot's streams, snapshots, and workspaces, initially sorted by stream name.

- *Graphical/tabular mode*: both of the above -- the tab is divided into a graphical pane (above) and a tabular pane (below). A movable separator bar enables you to adjust the space allocation to the panes.

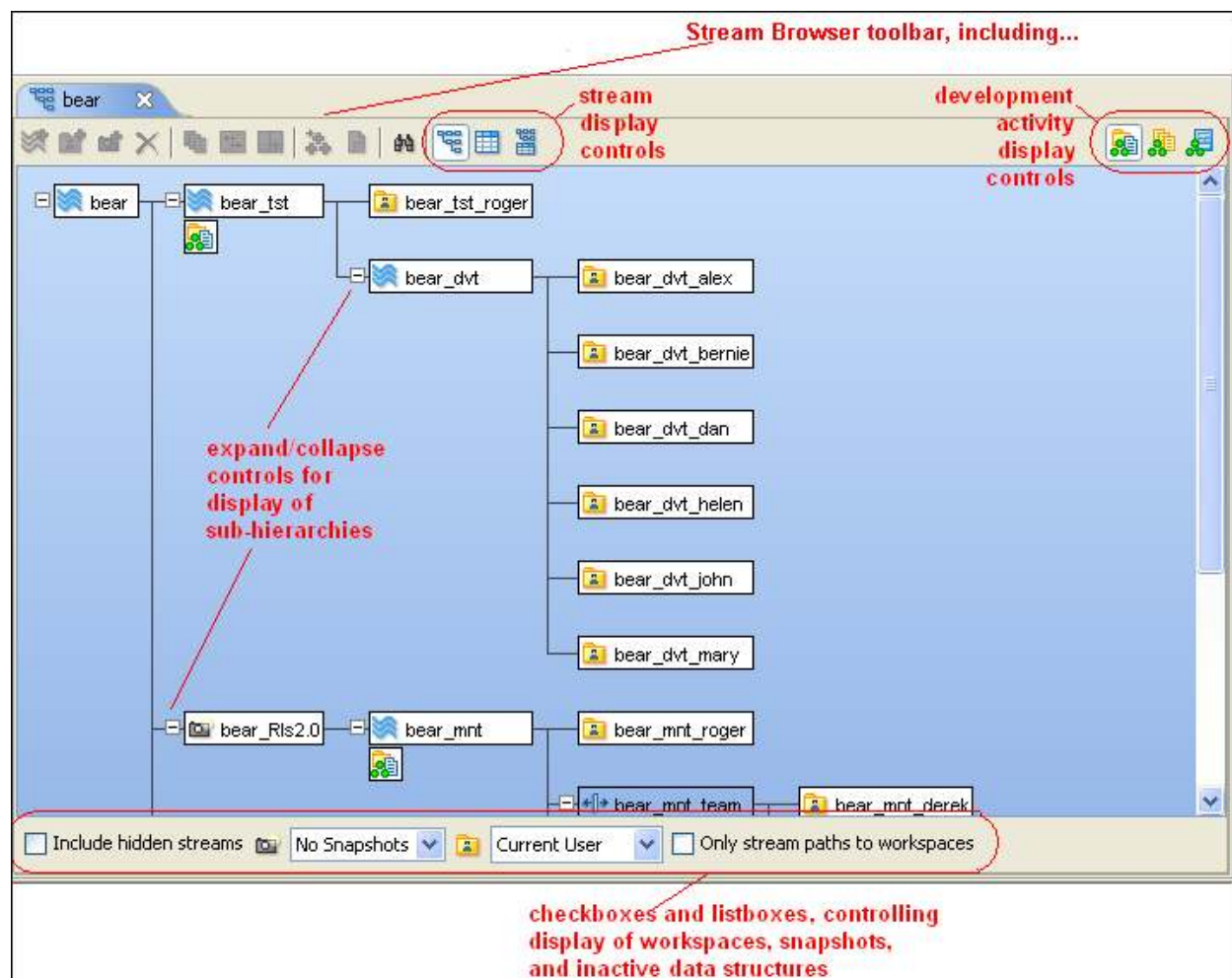
You can have any number of StreamBrowser tabs open at the same time, each for a different depot.








You can also open multiple StreamBrowser tabs on the *same* depot, by invoking the *Clone Tab* command on the context menu of the tab control.

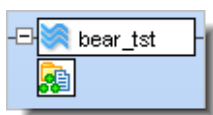
## Graphical StreamBrowser Display

The graphical display of a depot's stream hierarchy is organized as follows:



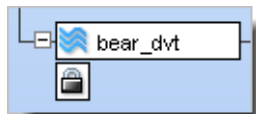
- The depot's root stream (top-level) is at the left edge.

- A given stream's children appear to its right; the children are arranged vertically, in this order:
  - oWorkspaces, in alphabetical order
  - oDynamic streams with no *basis time* and pass-through streams, in alphabetical order (capitalized stream names appear first, followed by stream names starting with the characters [ , \, ], ^, \_ , and ` , followed by non-capitalized stream name)
  - oDynamic streams with basis times and snapshots, ordered by basis time (most recent first)
- Each stream is annotated with an icon that indicates the kind of stream:
  -  *dynamic* ("normal") stream
  -  *time-based* dynamic stream
  -  *workspace stream*
  -  *snapshot* (static stream)
  -  *pass-through* stream
- Each stream or workspace that has *active* elements (that is, a non-empty *default group*) is



displayed with a special control that opens and closes a *subwindow showing the activity's details*. See [The Default Group of a Workspace or Stream](#) on page 135 for a general discussion of the default group.

- Each stream can have a *lock*, which prevents certain operations involving that stream from being performed.



If a lock exists on a stream, a special control appears, which provides access to [The Locks Dialog Box](#) on page 53.

- If you have set a *Stream Filter*, or limited the StreamBrowser display, only a subset of the streams in the depot appear. The Stream Filter is applied first; then the other *display restrictions* are used to further restrict the set of streams shown. Streams appearing in the filter are marked with a star (★) in graphical and tabular modes.

## Tabular StreamBrowser Display

In tabular display mode, the StreamBrowser tab contains a table listing some or all of the current depot's streams (including snapshots and workspaces). If you have used the global stream filter to restrict the display, only the stream paths and children (if any) for the selected streams appear in the table. (A stream path is a subset of the stream hierarchy that starts at the root stream and ends at the stream being referenced. )

The checkboxes and list box at the bottom of the tab work the same way as in graphical mode to control which streams are included in the table. The StreamBrowser's tabular mode offers the *same commands* (toolbar and context menus) as the graphical mode.

See also: [Working with Tables](#) on page 9

## Working in the StreamBrowser

The following sections describe working in the StreamBrowser. Note that many of these operations can be performed only in the graphical display, not in the tabular display.

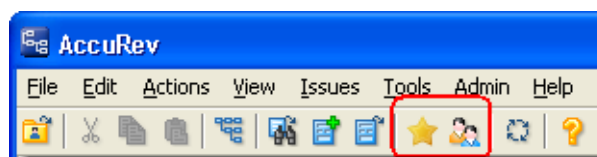
### Controlling the Display

Initially, the StreamBrowser displays all of the depot's currently active dynamic streams, along with workspaces for the current user. It does not display any snapshots, nor any item that you have deactivated with the *Remove* command.



*Note: If you have set a Stream Filter, only a subset of the streams in the depot appear.*

### Filtering Streams and Users/Groups

The main AccuRev toolbar contains two controls that can affect the StreamBrowser display.



The result of these filters will be applied to the data displayed in the StreamBrowser before adjustments made via the controls within the StreamBrowser.

- The *Filter Streams*  button allows you to set up a *stream filter* to limit the *stream hierarchy* shown in the StreamBrowser and any list of streams available in the AccuRev GUI. See [Quick Stream Filtering Options](#) on page 40 for more information.
- The *Filter Users/Groups*  button allows you to restrict the set of users and groups displayed in the StreamBrowser and other locations in the AccuRev GUI. See [Filter Users/Groups \(Tools > Filter > Users/Groups Command\)](#) on page 41 for more information.

### Streams, Snapshots, and Workspaces

You can adjust the numbers and types of streams shown by using the controls at the bottom of the StreamBrowser tab:



#### Include hidden streams

This checkbox is a toggle switch: it reveals/hides the data structures that have been *Remove'd*. (The *Remove* command doesn't actually delete anything from the depot; the data structure just becomes invisible and inactive.)

## Snapshots

This drop-down list controls how many snapshots are visible in the StreamBrowser. It contains the choices *No Snapshots* (the default), *All Snapshots*, *Past week*, *Past 2 weeks*, and *Past 4 weeks*. (Check the *Include hidden streams* checkbox if you want to see snapshots that have been *Remove'd*.) No matter what the setting, a snapshot remains visible if there are any workspaces or streams below it.

## Workspaces

This drop-down list controls which of the depot's workspaces are visible in the StreamBrowser. It contains the choices *All Workspaces*, *No Workspaces*, *Current User* (the default), *Unassigned*, and the name of each registered AccuRev user and group. (Check the *Include hidden streams* checkbox if you want to see workspaces that have been *Remove'd*.) To control which users and groups appear in this list, use the [Filter Users/Groups \(Tools > Filter > Users/Groups Command\)](#) on page 41 command.

## Only stream paths to workspaces

This checkbox is a toggle switch. Select it to filter the streams so that only the *stream paths* to workspaces -- the streams directly connecting the root stream to the workspace -- are visible in the StreamBrowser. Unselect it to remove the filter. Use the *Workspaces* list to control which workspaces are shown.

## Expanding and Collapsing Subtrees

In the graphical display mode, each stream that has "children" (workspaces and/or snapshots) is displayed with an expand/collapse control. Collapsing causes the entire hierarchy below the stream to disappear from the screen. This affects the StreamBrowser display only. It does not affect the operation of the stream in any way.

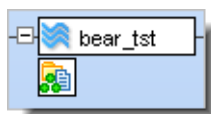
## Historical View of the Stream Hierarchy

If the *Enable StreamBrowser History* preference is checked ([AccuRev Preferences \(Tools > Preferences Command\)](#) on page 42), a set of *history controls* are added to the toolbar. This enables you to view the stream hierarchy as it existed at any point in the past.



## Viewing a Stream's Current Development Activity

The StreamBrowser can show the *development activity* currently taking place in each stream or workspace.



A control below the stream or workspace opens or closes a subwindow that displays the details of the development activity. The activity details can appear in several ways -- *by element*, *by transaction*, or *by issue record*.

Use the development-activity mode controls at the right side of the StreamBrowser toolbar to determine how the activity details will be displayed. (The icons on the controls below the streams and workspace change accordingly.) You can change modes either before or after opening an activity subwindow.

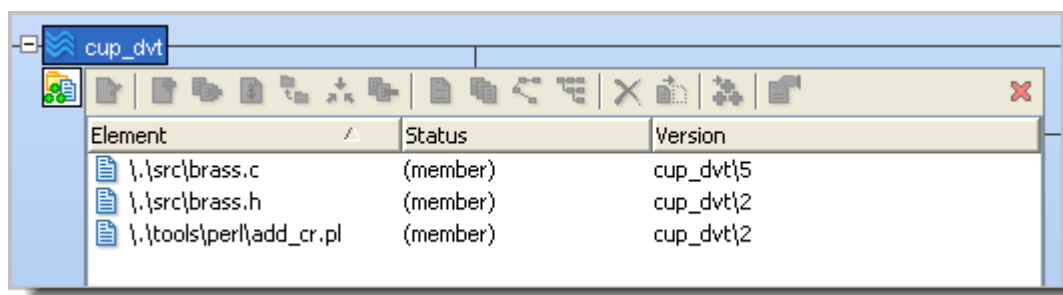


## Viewing Activity By Element

When displaying activity by element, the subwindow displays:

- *dynamic stream*: the results of a *Default Group* search
- *workspace*: the intersection of the results of a *Default Group* search and the results of a *Pending* search

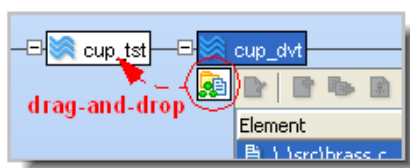
See [Active Versions vs. Pending Changes in a Workspace](#) on page 136 for more information on the relationship (and differences) between these two searches.



You can select one or more of the element versions and invoke commands on them, using the subwindow's toolbar. The commands available are the same as in the File Browser: see [Details Pane Command Reference](#) on page 80.

You can also perform these drag-and-drop operations:

- Dragging the subwindow control (but not elements within the subwindow) to the parent stream or another dynamic stream performs a *Promote* or *Send to Change Palette* operation on all the element versions in the subwindow.



- Dragging the subwindow control (or a selection of one or more versions within the subwindow) to a workspace performs a *Send to Workspace* operation.

### Viewing Activity By Transaction

When displaying a stream's activity by element, the subwindow displays the *transactions* that created the versions that are currently active in the stream. This display is the same as that in the Summary pane of an Active Transaction tab; see [The Active Transactions Tab \(Show Active Transactions command\)](#) on page 168. You can select one or more transactions and perform commands on them, just as in the Active Transactions tab.

### Viewing Activity By Issue Record

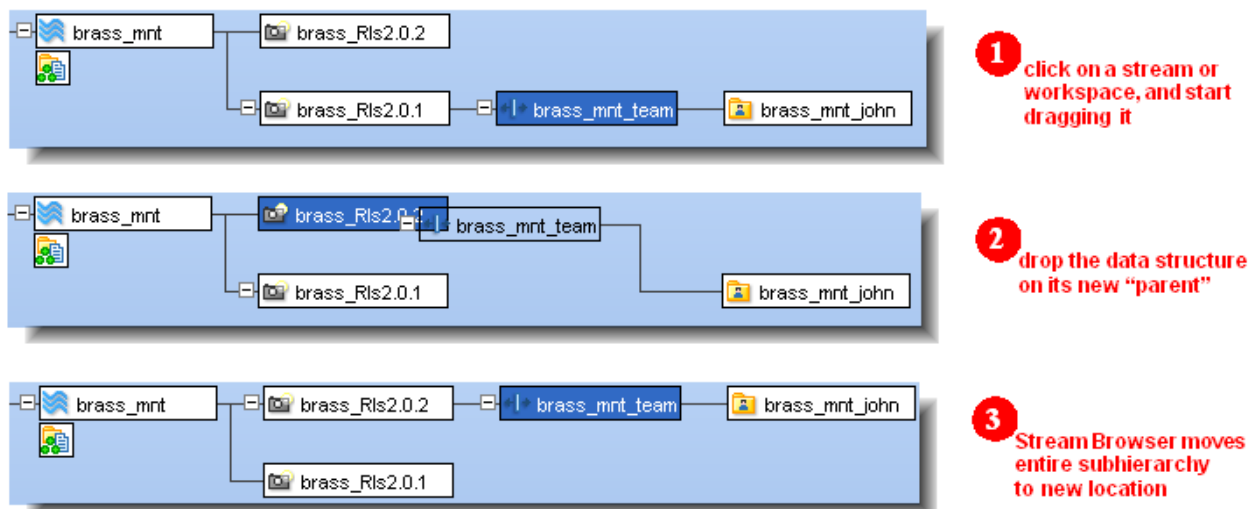
When displaying a stream's activity by issue record, the subwindow displays the *issue records* that are in a particular stream. This display is the same as that in the Issues pane of a Stream Issues tab (see [The Stream Issues and Stream Diff \(Issues\) Tabs](#) on page 157). You can select one or more issue records and perform commands on them, just as in the Stream Issues tab.

You can also perform this drag-and-drop operation:

- Dragging the subwindow control (but not elements within the subwindow) to the parent stream or another dynamic stream performs a *Promote* or *Send to Change Palette* operation on all the versions in the *change packages* of the subwindow's issue records.

## Reparenting Streams and Workspaces

AccuRev lets you change the backing stream (parent stream) of any dynamic stream or workspace. The StreamBrowser makes it simple: you just drag-and-drop a stream or workspace from its current location in the hierarchy to its new parent.



The entire subhierarchy moves to the new location. Changing a data structure's location in the stream hierarchy is called **reparenting**. If the StreamBrowser display contains many streams or the stream structure is very complex, an alternative method of reparenting a stream is to select *Reparent* from the context menu of the stream or workspace, then navigate to and click the new parent stream. You are asked to confirm the operation.

Important: After you change the location of a workspace, be sure to Update it. This ensures that the workspace contains the correct set of versions, many of which it will inherit from its new parent. Likewise, after changing the location of a stream, all workspaces in the subhierarchy below that stream should be Update'd.

### Notes:

#### Snapshots

You cannot reparent a snapshot; both the contents and the parentage of snapshot are fixed permanently.

#### Exclusive file locking workspaces

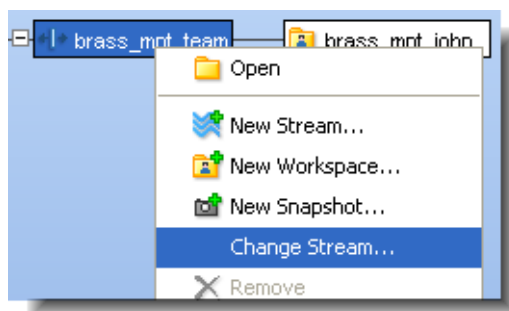
You cannot reparent (or change any of the other specifications) of a workspace created with the *exclusive file locking* feature.

#### Time-based streams

If you reparent a *time-based stream*, AccuRev asks whether you wish to retain the basis time or remove it.

## Reconfiguring a Stream

The drag-and-drop operation changes one property of a stream: its parent. You can also change a stream's properties using the *Change Stream* command on its context menu.



Although you can give a new name to an existing stream, you cannot proceed to create a new stream with the old name. The old name remains associated with its stream. The only way to reuse a stream name is to completely remove the stream's depot from the AccuRev repository, using the AccuRev administration utility, *maintain*.

## Setting or Changing a Stream's Basis Time

By default, a stream inherits all versions from its parent, no matter when those versions were created. If you assign a *basis time* to a stream, it inherits only those versions created before the specified point in time. (The File Browser's *include/exclude* facility provides another way to restrict which versions get inherited from the parent stream.)

## Creating New Streams, Snapshots, and Workspaces

The context menu of any data structure in the StreamBrowser includes commands for creating new structures at that point in the hierarchy. *New Stream* and *New Snapshot* display dialogs similar to that of *Change Stream*. The *New Workspace* command invokes a wizard that steps you through the process of defining a new workspace: you specify a name and a location on disk; you can also make some optional settings, such as controlling how line endings in text files are to be handled.

## Commands Available in a StreamBrowser Tab

### Open

Using a *File Browser* tab, display the entire contents of the selected stream, snapshot, or workspace.

### New Stream

Create a new dynamic stream or pass-through stream, based on (that is, as a child of) the selected stream or snapshot. A *Stream Configuration* dialog appears.

### New Workspace

Create a new workspace, based on (that is, as a child of) the selected stream or snapshot. A *New Workspace* wizard helps you to configure the workspace.

### New Snapshot

Create a new snapshot, based on (that is, as a child of) the selected stream. A *New Snapshot* dialog appears.

### Change Stream

Modify the configuration of the selected dynamic stream or pass-through stream. A *Stream Configuration* dialog appears.

### Reparent

*Change the backing stream* (parent stream) of the selected dynamic stream or workspace.

### Remove

After *confirmation*, hide the selected stream, snapshot, or workspace. This removes the object from the StreamBrowser display.

Removal affects how the object can be used:

- Workspace: deactivates the workspace, so that it cannot be used in AccuRev commands.
- Stream or snapshot: no effect; you can still promote to the stream, create a substream of it, or reparent it.

*Note: Removing a workspace that belongs to another user*

You cannot do this using the AccuRev GUI. The *rmws -s* command in the AccuRev CLI provides this capability.

### Reactivate

Undo a *Remove* command for the selected stream, snapshot, or workspace.

### Locks ... (Lock/Unlock Stream)

(Dynamic stream only) Lock a stream, to disable various operations that modify the stream. If a stream is already locked, a lock icon appears in the StreamBrowser graphical display, and in the Locks column of the tabular display. A Lock dialog appears: see [The Locks Command](#) on page 53.

### WIP (Work in Progress)

Open a *WIP* tab, displaying the active elements in workspaces and/or streams related to the selected stream.

### Search for Stream




Find the stream(s) whose name includes a specified character string. A *Search for Stream* dialog appears.

### Show All Overlaps

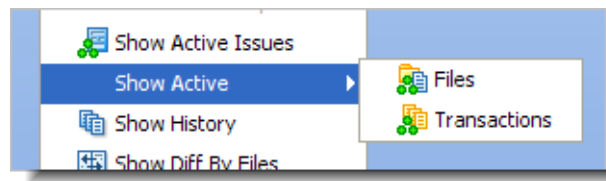
(Workspace only) Using an *Overlaps* tab: for active file elements (those in the workspace's default group), display all overlaps involving versions in the workspace's entire backing chain (higher-level streams) -- not just overlaps between the workspace and its immediate parent stream.

### Show Active ...

There are three commands to display the stream's development activity, using the appropriate tool for the current *development-activity mode*:

-  Element mode: open a *File Browser* tab on the selected stream, and execute a search for active elements.
-  Transaction mode: open an *Active Transactions* tab, showing the transactions that created the versions currently in the selected stream's default group.
-  Issue Record mode: open a *Stream Issues* tab, showing the change packages that have been (partially or completely) incorporated into the selected stream.

The way the three commands appear on a stream's context menu varies with the development-activity mode: the command for the current mode appears at the top-level of the context menu; the commands for the other two modes are in a submenu. For example, the following illustration shows how the menu choices appear if the StreamBrowser is currently displaying activity by issue record.



### Show History

Using a *History Browser*, display the transactions that created versions in the selected stream.

### Show Diff By Files

Using a Diff Streams window, show the element-by-element differences between the selected stream and another stream (which AccuRev prompts you to specify).

### Show Diff By Issues

Using a *Stream Issues* window, show the differences between the selected stream and another stream (which AccuRev prompts you to specify). The differences are shown in terms of change packages instead of individual elements.

### Send to Change Palette

Send the active elements for the selected stream to the *Change Palette*.

### Show Patch List

Using a *Patch List* tab, list all the individual versions whose changes are in the selected stream, but are not in another stream (which AccuRev prompts you to specify).

### Add to Stream Filter / Remove from Stream Filter

If the selected stream is not in the *stream filter*, add it and immediately apply the filter. If the selected stream is in the stream filter, remove it and immediately update the filter. See [Quick Stream Filtering Options](#) on page 40 for more information on editing and using the stream filter.

### Graphical Display

Display the depot's stream hierarchy as a tree.

#### **Tabular Display**

Display the depot's stream hierarchy as a table.

#### **Graphical/Tabular Display**

Display the depot's stream hierarchy as a tree in one pane, and as a table in another pane.

## The Overlaps Tab

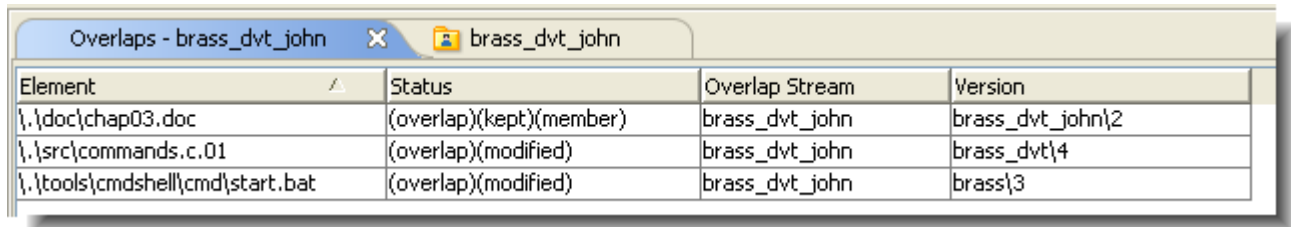
The Overlaps tab displays all versions with **(overlap)** status, both in the selected workspace and in the workspace's entire *backing chain*.

### Opening an Overlaps Tab

Choose *Show All Overlaps* from the context menu of a workspace on a *Stream Browser* tab.

### Overlaps Tab Layout

The Overlaps tab contains a table with these columns:



Element	Status	Overlap Stream	Version
\\.\doc\chap03.doc	(overlap)(kept)(member)	brass_dvt_john	brass_dvt_john\2
\\.\src\commands.c.01	(overlap)(modified)	brass_dvt_john	brass_dvt\4
\\.\tools\cmdshell\cmd\start.bat	(overlap)(modified)	brass_dvt_john	brass\3

#### **Element (or separate "Name" and "In Folder" columns)**

The pathname within the depot of the element with the overlap.

#### **Status**

The status of the version in that workspace or stream.

#### **Overlap Stream**

The workspace or stream that contains a version with **(overlap)** status.

#### **Version**

The *version-ID* of the version in that workspace or stream.

### Working in an Overlaps Tab

For versions in dynamic streams (but not workspaces), you can begin the process of resolving the overlap:

1. Select one or more versions located in the same dynamic stream.
2. Right-click the selection, and choose *Send to Change Palette* from the context menu.

This opens a *Change Palette* tab, with an entry for each version you selected. The destination stream is automatically set to the parent of the stream where those versions are located. You can resolve the overlaps by performing *merges* on those versions.

# The Remove Stream Command

## (and the Remove Workspace Command)

You can invoke the *Remove* command on a stream, snapshot, or workspace. This hides the item from the *Stream Browser* and the *Workspaces* tab, but you can use the checkbox in both these tabs to view removed ("hidden") items.

This command does not remove the object from the depot — that would violate AccuRev's *TimeSafe* property. It disables some operations, but not all:

- You cannot work in a removed workspace.
- You can promote versions to a removed stream.

To bring back a removed object, make it visible with the *Show including hidden* checkbox, then invoke the *Reactivate* command on it.

## Invoking the Remove Stream Command

In the *StreamBrowser*, choose *Remove* from the context menu of a stream or snapshot.

Restrictions:

- This command is disabled if there are any streams, snapshots, or workspaces below the one to be removed.

## Invoking the Remove Workspace Command


In the *StreamBrowser*, choose *Remove* from the context menu of a workspace. Restriction: You must be logged in as the user who owns the workspace


Also see: [Filter Streams \(Tools > Filter > Streams Command\)](#) on page 37.

# Default Group Subwindows in the StreamBrowser

Each workspace and dynamic stream has a **default group**, consisting of the elements that are currently active in that context. (More precisely, the default group contains a *version* of each active element.) In graphical mode, the *StreamBrowser* can display the default group for one workspace or stream at a time.

## Opening a Default Group Subwindow

The *StreamBrowser* displays a  Default Group icon for each workspace and dynamic stream with a non-empty default group. Click this icon to open a subwindow displaying the default group.

You can display default groups just one-at-a-time -- the subwindow closes automatically if you click the  icon of another workspace or stream.

*Note: If you don't see any  icons, but you know some elements are active...*

A user preference (*Display Default Group in StreamBrowser*) controls whether the Stream Browser spends the time to determine the contents of the default group of each stream and workspace in its display.


## Default Group Subwindow Layout

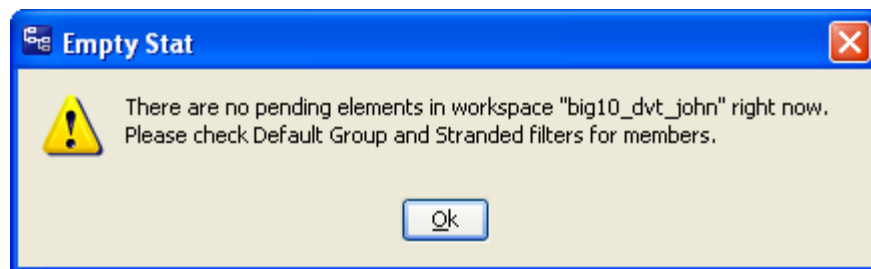
The default group subwindow has essentially the same layout as the File Browser's *Details pane*:

- A table, showing the members in the default group.
- A toolbar, for performing operations on the members.

## StreamBrowser's default group subwindow vs. File Browser's Default Group search


The results of a Default Group search in the File Browser are not always the same as the contents of the StreamBrowser's Default Group subwindow:

- The File Browser's search displays all the members of the default group.
- The StreamBrowser's subwindow displays only those members with Pending status. This does not include elements that are active by virtue of a *Send to Workspace* or *Anchor* command. If the default group is non-empty, but contains *only* non-Pending elements, the  icon will appear, but clicking it brings up an error box.



- Working in a Default Group Subwindow

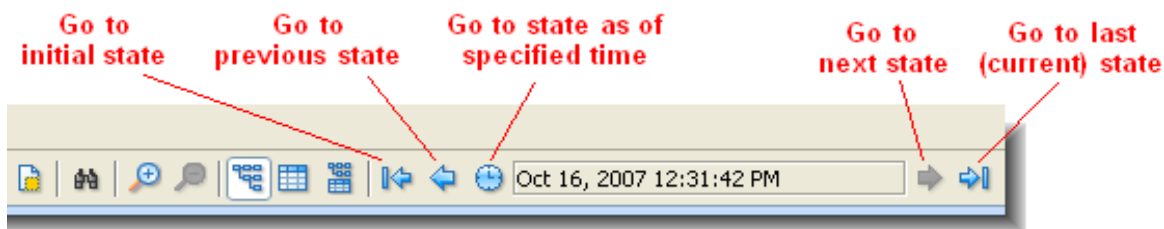
The toolbar in the Default Group subwindow contains many of the same buttons as the toolbar of the *File Browser's* Details pane. In particular, you can select one or more elements, then *promote* them to the *backing stream*.

Note: If you want to promote all the elements in the default group subwindow to the backing stream, you don't need to open the subwindow at all. Just drag-and-drop the  icon below the workspace or stream to the backing stream. (The backing stream is the only stream you can target with this drag-and-drop operation.)

## StreamBrowser History

Like the individual files under version control, a depot's *stream hierarchy* undergoes change, too. Initially, the depot contains a single *base stream*. Thereafter, additional dynamic streams are created, along with snapshots and user workspaces. Each such change in the stream hierarchy

creates a new "state" of the hierarchy. You can browse through all these states, like browsing through a slide show, using the History controls in the StreamBrowser toolbar:




## Find Stream

The StreamBrowser search feature enables you to find a stream or workspace whose name contains a specified character string. You can use this feature in any of the StreamBrowser's display modes: *graphical*, *tabular*, or *combined*.

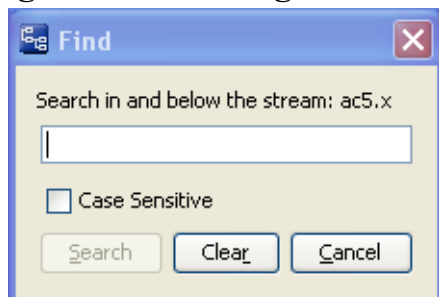
### Invoking the Find Stream Dialog

You can access the StreamBrowser Find dialog in one of three ways:

- Click the  icon in the StreamBrowser toolbar.
- Type Ctrl-F in a StreamBrowser view.
- Right-click over a stream and click Find.

Note that if you right-click, your search will be limited only to the streams under the stream where you right-clicked. Otherwise, ALL streams in the depot matching your search will be found.

### Using the Find Dialog



Type characters in the text box that match all or part of the stream name(s) you are looking for. Make sure the *Case Sensitive* option is set as desired. You can enter any character that is legal for an AccuRev stream name. Wildcards ("\*" and "?") are accepted. The find operation will match the string in any part of the name. So for example, performing a case-insensitive search for "Devel" will return streams with names like "devel\_stream", "Old\_Development".

## Displaying and Filtering Stream Find Results

The results are displayed in a StreamBrowser tabular display.


Search For: "te?t" on depot2 Stream; Showing 5 Results

Search Again...







Clear Search

Filter Results:

Enter text



Reset Filter

Name	Base	Type	Basis Time	#	Changes	Locks	Stream Filter
 DOC_tmp_test_dir_rvrt_bfrederick	...	Workspace		...			
 i18n_test	...	Normal		...			
 isaac_test	...	Normal	 Dec 8, 2011 2:24:28 PM	...			
 ls_debug_test1	...	Normal		...			
 Test Utilities	...	Normal		...			

To fine-tune the results, you can enter characters in the Filter Results field. As you type a string, the streams shown in the results display will be narrowed down to only those containing the characters you enter. The rules for Filter strings are the same as for Find String.

## The Stream Diff (Files) Tab



### (Show Diff By Files command)

The Stream Diff (Files) tab displays the element-by-element differences between two streams. Each stream can be a *dynamic stream*, a *snapshot*, or a *workspace stream*. AccuRev determines the differences by examining the elements' *version-IDs* in each stream. It does not compare the contents of file versions (but you can, using this tab's *Show Difference* command).


A stream comparison can report these differences

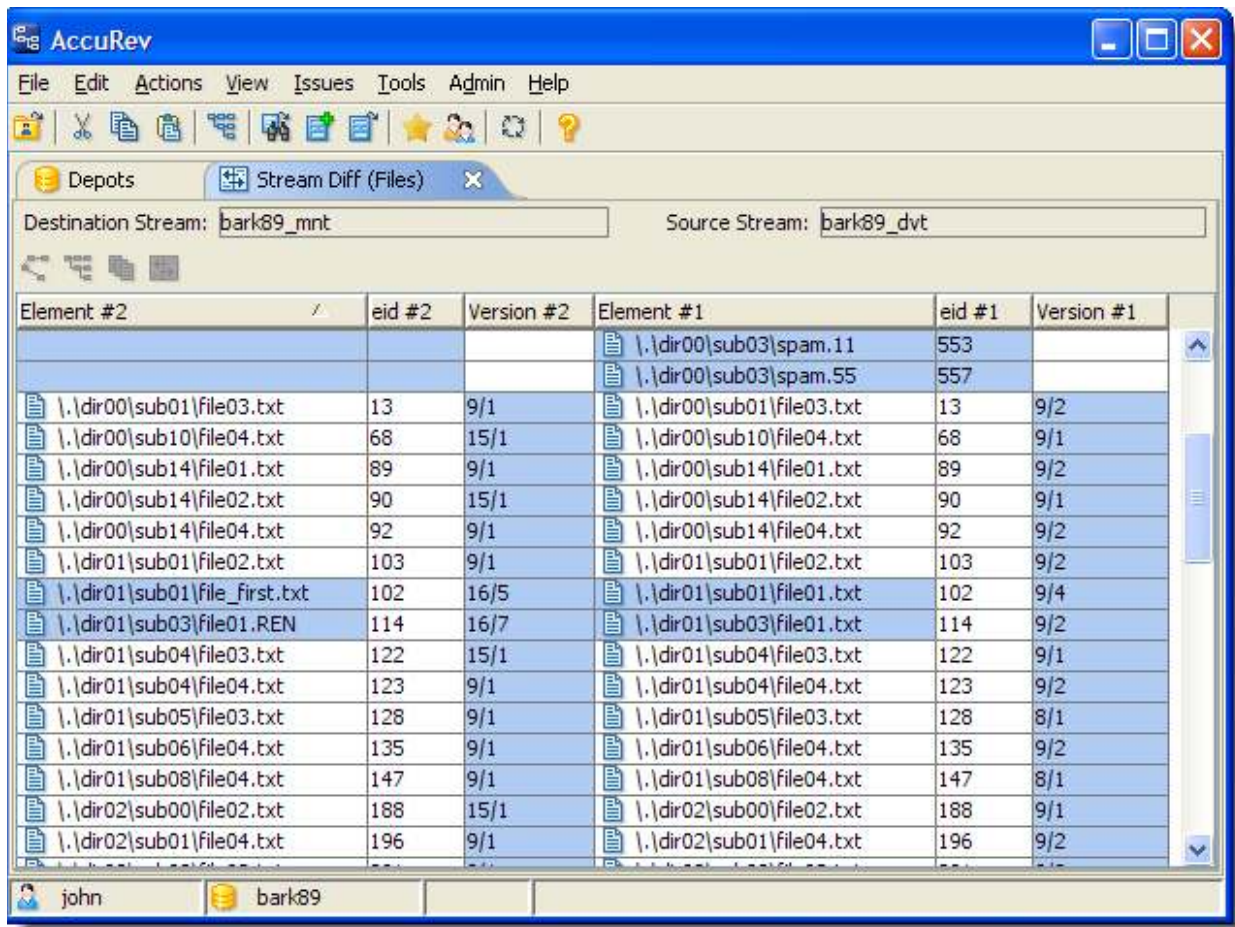
- A file element has been modified in one or both streams. (When analyzing the contents of a workspace, this command always uses the version currently in the workspace stream (which might be inherited from a higher-level stream). This version differs from the file in the workspace tree if the file has **(modified)** status.)
- An element has been renamed in one or both streams.
- A new element has been added to one of the workspaces (*Add to Depot* command) or streams (*Promote* command)
- An element has been removed from one of the streams (*Defunct* command).

## Opening a Stream Diff (Files) Tab

1. In the StreamBrowser, select a stream, snapshot, or workspace.
2. Choose the *Show Diff by Files* command from the selection's context menu, or click the  button on the StreamBrowser's toolbar.
3. The mouse pointer changes to . Left-click another stream, snapshot, or workspace.

## Stream Diff (Files) Tab Layout

The Stream Diff (Files) tab contains a table, each row of which details one element. Above the table is the name of the **Destination Stream** (the one you selected with the  mouse pointer) and **Source Stream** (the one you selected before invoking the command).



Element #2	eid #2	Version #2	Element #1	eid #1	Version #1
			\\.\dir00\sub03\spam.11	553	
			\\.\dir00\sub03\spam.55	557	
\\.\dir00\sub01\file03.txt	13	9/1	\\.\dir00\sub01\file03.txt	13	9/2
\\.\dir00\sub10\file04.txt	68	15/1	\\.\dir00\sub10\file04.txt	68	9/1
\\.\dir00\sub14\file01.txt	89	9/1	\\.\dir00\sub14\file01.txt	89	9/2
\\.\dir00\sub14\file02.txt	90	15/1	\\.\dir00\sub14\file02.txt	90	9/1
\\.\dir00\sub14\file04.txt	92	9/1	\\.\dir00\sub14\file04.txt	92	9/2
\\.\dir01\sub01\file02.txt	103	9/1	\\.\dir01\sub01\file02.txt	103	9/2
\\.\dir01\sub01\file_first.txt	102	16/5	\\.\dir01\sub01\file01.txt	102	9/4
\\.\dir01\sub03\file01.REN	114	16/7	\\.\dir01\sub03\file01.txt	114	9/2
\\.\dir01\sub04\file03.txt	122	15/1	\\.\dir01\sub04\file03.txt	122	9/1
\\.\dir01\sub04\file04.txt	123	9/1	\\.\dir01\sub04\file04.txt	123	9/2
\\.\dir01\sub05\file03.txt	128	9/1	\\.\dir01\sub05\file03.txt	128	8/1
\\.\dir01\sub06\file04.txt	135	9/1	\\.\dir01\sub06\file04.txt	135	9/2
\\.\dir01\sub08\file04.txt	147	9/1	\\.\dir01\sub08\file04.txt	147	8/1
\\.\dir02\sub00\file02.txt	188	15/1	\\.\dir02\sub00\file02.txt	188	9/1
\\.\dir02\sub01\file04.txt	196	9/1	\\.\dir02\sub01\file04.txt	196	9/2

The columns of this table are:

#### Element #2 (or two columns: Name #2, In Folder #2)

The pathname of the same element in the destination stream. If empty, the element is present only in the source stream. Note that a *user preference* controls whether the element pathnames are displayed in a single column, or in two columns.

#### eid #2

The *element-ID* of the element.

#### Version #2

The *version-ID* of the destination stream's version of the element.

#### Element #1 (or two columns: Name #1, In Folder #1)

The pathname of the element in the source stream. If empty, the element is present only in the destination stream. Note that a *user preference* controls whether the element pathnames are displayed in a single column, or in two columns.

#### eid #1











Same as eid #2. (The element-ID is immutable. It is unique within the depot, but not within the entire repository.)

## Version #1

The *version-ID* of the source stream's version of the element.

## Working in a Stream Diff (Files) Tab

In any row of the Stream Diff table, you can left-click to select an element's version in either stream/workspace:

Element #1	eid #1	Version #1	Element #2	eid #2	Version #2
 \.\src\brass.c.01	8	5/4	 \.\src\brass.c	8	4/3
 \.\src\brass.h.01	9	5/8	 \.\src\brass.h	9	11/1
 \.\doc\chap01.doc	3	5/5	 \.\doc\chap01.doc	3	11/1
 \.\doc\chap02.doc.01	4	5/7	 \.\doc\chap02.doc	4	11/2
 \.\doc\chap03.doc	5	5/7	 \.\doc\chap03.doc	5	11/1

Then, you can invoke a command, using the Stream Diff tab toolbar or the selection's context menu. Some commands operate on the particular version you selected; other commands operate on the overall element.

## Commands That Operate on the Selected Version

### Open (equivalent to double-click)

**Windows:** Run the appropriate command on the file, according to its file type. (The Windows file-typing system — "file associations" — does not provide for assigning a file type if the filename has no suffix.)

**UNIX:** Open a text editor on the file.

### View

(text files only) Open a text editor on a temporary copy of the selected version.

### Save As

Copy the selected version to another filename.

## Commands That Operate on the Overall Element

### Browse Versions

Open a *Version Browser* tab, showing all the versions of the selected element, and their interrelationships (ancestry).

### Browse Stream Versions

Open a *Stream Version Browser* tab for the selected element. This tab displays the depot's stream hierarchy, much like the *Stream Browser*. But on this tab, each stream represents the *version* of the specified element that appears in that stream.

### Show History

Open a *History Browser* tab, containing the transactions involving the selected element.

### Show Difference

(file elements only) Compares the contents of the two streams' versions of the selected element. Note that when analyzing the contents of a workspace, this command always uses the version currently in the workspace stream (which might be inherited from a higher-level stream). This version differs from the file in the workspace tree if the file has **(modified)** status.


## The Stream Issues and Stream Diff (Issues) Tabs

The Stream Issues and Stream Diff (Issues) tabs are essentially similar. A table in the Issues pane (upper) displays the *issue records* that are in a particular stream. The Change Package Contents pane (lower) displays the entries in a selected issue record's *change package*.



*Note: Terminology -- issues "in" a stream*

Strictly speaking, the only objects that are "in" a stream are versions. It makes sense to describe a set of versions as being "in" a stream. And so, it makes sense to describe the set of versions in a change package as being "in" a stream. From there, we make the leap to describing the AccuWork issue record containing the change package as being "in" a stream.

### Opening the Tab

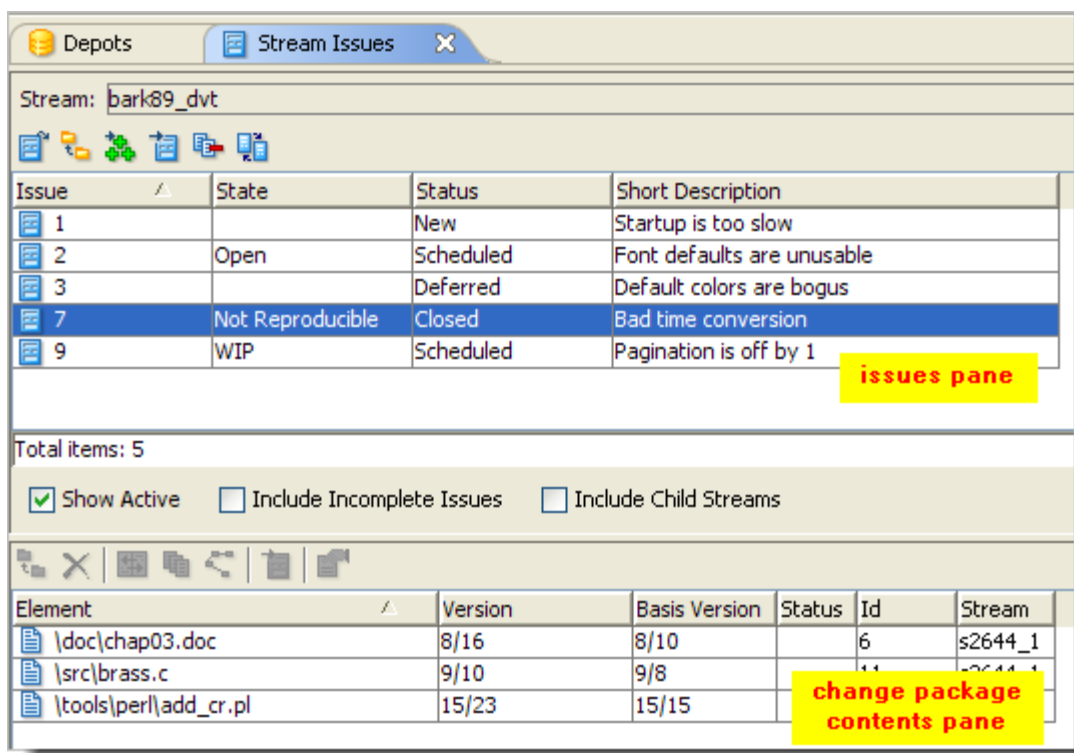
- To open a Stream Issues tab, showing the contents of a stream in terms of change packages: choose Show Active Issues from the context menu of any stream, or snapshot, or workspace in the StreamBrowser. Note: If the StreamBrowser is displaying development activity by issue record, you can click the  icon under the stream itself to open a subwindow containing the same data as a separate Stream Issues tab.



- To open a Stream Diff (Issues) tab, showing the difference between two streams in terms of change packages:
  - In the StreamBrowser, select a stream, snapshot, or workspace.
  - Choose the Show Diff by Issues command from the selection's context menu, or click the  button on the StreamBrowser's toolbar.
  - The mouse pointer changes to  . Left-click another stream, snapshot, or workspace.

### Stream Issues Tab Layout

The Stream Issues tab includes two panes, each with its own toolbar:



- The Issues pane displays selected fields from the issue records whose change packages are "in the stream"—partially or completely. (To get here, use the Admin> Schema Editor command to open a Schema Editor tab. Go to the Change Packages subtab, and select fields to be displayed in the upper Change Package Results pane). When you're comparing the contents of two streams, this pane displays the issues whose change packages are in one stream, but not the other.
- The Change Package Contents pane displays the entries in the **change package** of the currently selected issue record.

## Working in the Issues Pane

You can control which issues are displayed, and you can invoke commands on one or more issues' change packages.

### Controlling Which Issues are Displayed

Checkboxes at the bottom of the Issues pane control which issues are displayed. The *Show Active*, *Include Incomplete Issues*, and *Include Hierarchy* checkboxes appear only when you're displaying a single stream's change packages. (When comparing streams, AccuRev always proceeds as if these checkboxes are cleared.) The *Bidirectional* checkbox appears only when you're comparing two streams.

#### Show Active

If checked, an issue is included only if one or more of its versions was promoted *to* the stream, but has not yet been promoted *from* the stream to its parent. This helps you to concentrate on current programming efforts, rather than those that were completed long ago.

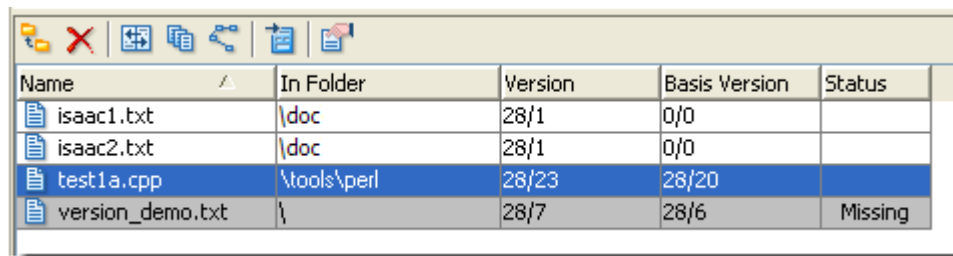
Note: For the purposes of the Show Active filter, a Revert to Backed operation has the same effect as a Promote operation. For a workspace, substitute the phrase "created in" for "promoted to".

### Include **Incomplete** Issues

If cleared, the listing includes only issues that are completely in the stream, where “completely in” means that for every change package entry, the head version (Version column) itself is in the stream, or one of its descendants is in the stream.

If checked, the listing includes all issues, including ones that are partially in the stream, where “partially in” means that for some but not all change package entries, the head version (Version column) itself is in the stream, or one of its descendants is in the stream.

On the Change Package Contents pane, the Status column and background shading indicate which change package entries are "in" the stream and which are not in ("missing" from) the stream.



Name	In Folder	Version	Basis Version	Status
isaac1.txt	\doc	28/1	0/0	
isaac2.txt	\doc	28/1	0/0	
test1a.cpp	\tools\perl	28/23	28/20	
version_demo.txt	\	28/7	28/6	Missing

### Include **Hierarchy**

(available only if *Show Active* is checked) If checked, also includes issues that are in the streams below the selected stream.

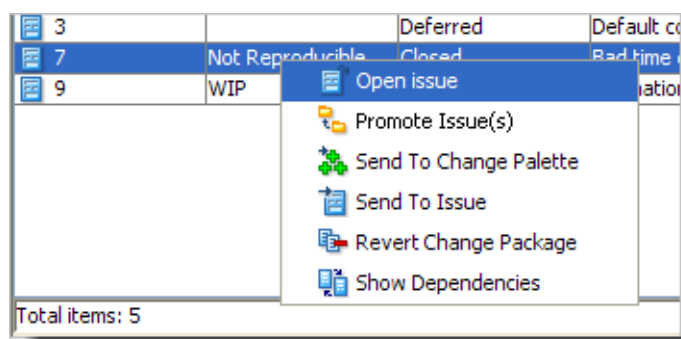
### Bidirectional

(Stream Diff only) If cleared, the Issues pane displays the issue records in the first stream ("source"), but not in the second stream ("destination").

If checked, the listing also displays the issues that are in the second stream, but not in the first stream. Color-shading helps you to distinguish the two sets of change packages.

### Commands Available in the Issues Pane

Select one or more issues to invoke commands on the issues, or on their change packages.



These commands are available through context menus and in the toolbar of the issues pane.

## Open Issue

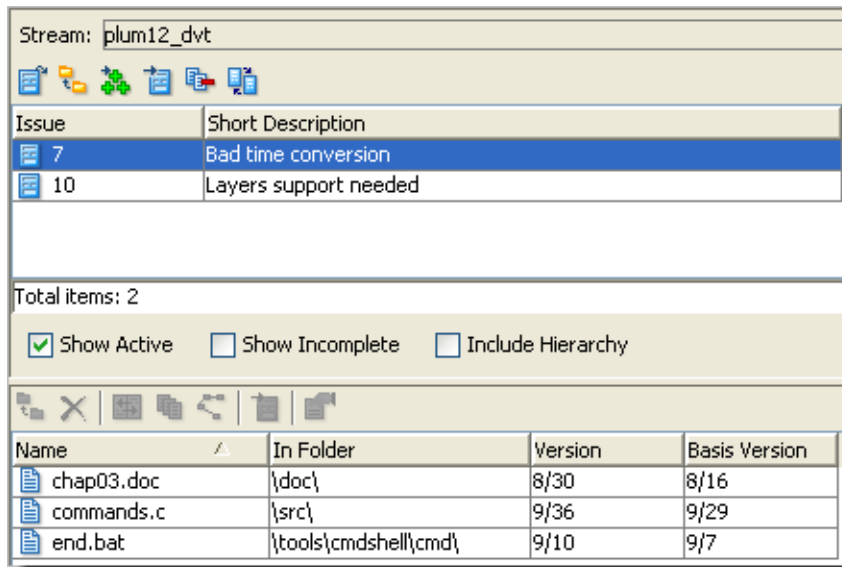
Open an edit form on the selected issue, and display the Changes tab.

## Promote Issue(s)

**Stream Issues tab** (enabled only if *Show Active* is set and *Show Incomplete* is cleared): promotes each currently-active head version in the issue's change package to the parent stream. That is, if a version listed in the Version column is currently active in the stream on which you invoked the *Show Issues* command, that version is promoted to the parent stream.

### *Example:*

The following illustration shows the Stream Issues tab for dynamic stream *plum12\_dvt*.



Issue	Short Description
7	Bad time conversion
10	Layers support needed

Total items: 2

☒ Show Active   ☐ Show Incomplete   ☐ Include Hierarchy

Name	In Folder	Version	Basis Version
chap03.doc	\doc\	8/30	8/16
commands.c	\src\	9/36	9/29
end.bat	\tools\cmdshell\cmd\	9/10	9/7

Invoking the *Promote* command on issue #7 causes some or all these versions to be promoted. The versions promoted are the ones that are currently in the workspace's default group:

- version 8/30 of chap03.doc
- version 9/36 of commands.c
- version 9/10 of end.bat

**Stream Diff (Issues) tab** (works only for issues in the source stream, not for issues in the destination stream): promotes each currently-active head version in the issue's change package from the source stream to the destination stream. If there is an *overlap* between the two streams for any element, the *promote* operation fails with a "merge required" error box.

## Send to Change Palette

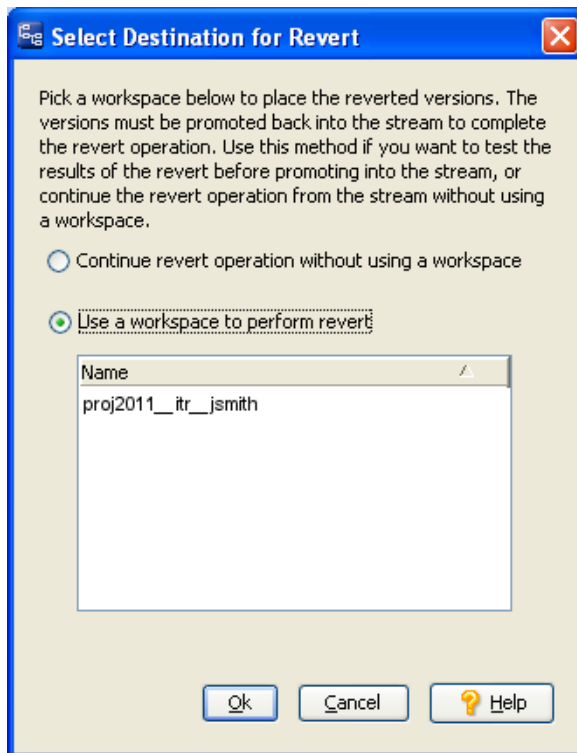
(enabled only if *Show Active* is set and *Show Incomplete* is cleared) Opens a Change Palette tab, containing each head version in the issue's change package.

## Send to Issue

Adds (or combines) the selected issue record's change package entries with those of one or more other issue records. A popup window prompts you to specify the destination issue record(s).

### **Revert Change Package**

(Formerly named *Revert by Change Package*.) Removes the changes in the selected issue's change package from the stream. A dialog appears offering you two options for performing this operation:



In reverse order, these options are:

- Use a workspace to perform revert — Choose this option if you wish to test the results of the revert before promoting them back into the stream. You must choose a workspace to which you have access to perform the revert. (All workspace appear in the selection list. If you select a workspace to which you do not have access, AccuRev will display an error). The revert operation typically involves one or more **reverse patch** operations on file elements' contents, performed with the Merge tool.

Important: If no valid workspaces exist, this dialog is bypassed and you are prompted whether or not to proceed without a workspace (see below). A "valid" workspace is one of your own workspaces which has local (or accessible mounted) storage on the machine that the revert is being performed from. If none of your workspaces are attached to the stream the issue is being reverted from, or for some reason are not accessible from the machine that the client is running on, you can still proceed with the revert, keeping the new file versions directly in the stream. However, AccuRev warns you that this is what happening if you choose to proceed with the revert operation.

Note: This feature is not intended to be used within a workspace; it is meant to be executed within a stream, but with the aid of a workspace to perform any necessary Merge operations. If you try to execute this operation directly within a workspace, you will get an error message: "You need to have at least one workspace associated with stream..." Try the following workaround:

1. Create a new stream between the workspace and its backing stream.
  2. Promote the issue to the new stream.
  3. Invoke the Revert Change Package Using Workspace command on that issue in the new stream.
- Continue revert operation without using a workspace — Similar to Use a workspace to perform revert, except that the results are placed directly in the stream, without the requirement of a workspace. This means that the results of the revert will be immediately propagated downstream in the hierarchy.

AccuRev prompts you for an issue number against which to record the Revert changes. Either create a new issue, or select an existing issue that you created specifically for this Revert. (You should rarely—if ever—select an existing issue that has been used for previous operations.) Then perform whatever merge changes are required by the revert.

Important: If no workspaces to which you have access exist, the dialog described above is bypassed and you are simply prompted whether or not to proceed without a workspace.

Note: Continue revert operation without using a workspace saves the step of merging and testing the revert in a workspace and then promoting the changes, but at the risk of propagating the changes as soon as the revert has been completed. This feature should not be used casually by developers, as you could easily propagate complex, untested changes to other users. This feature is useful for reverting relatively simple changes, or for backing out a change in an isolated build stream. AccuRev recommends that you build and test these changes in an updated workspace to verify the results, before promoting or cross-promoting the changes to other users.

### **Show Dependencies**

Opens an *Issue Dependencies* tab, displaying the change package dependencies of the selected issue record(s).

## **Working in the Change Package Contents Pane**

Select any change package entry and invoke any of the commands described below. In most cases, the operation is performed on the entry's head version (**Version** column). You can invoke the *Remove* and *Send to Issue* commands on a selection consisting of two or more change package entries.

### **Open (equivalent to double-click)**

**Windows:** Run the appropriate command on the file, according to its file type. (The Windows file-typing system — "file associations" — does not provide for assigning a file type if the filename has no suffix.)

**UNIX:** Open a text editor on the file.

### **View**

Open a text editor on a temporary copy of the currently selected version (text files only).

## Save As

Copy the currently selected version to another filename.

## Promote

Promote the head version(s) of the selected entry(s) to the parent stream.

## Remove

Remove the selected entry(s) from the change package.

## Diff Against Basis

Compare the selected version with the corresponding basis version.

## Show History

Open a *History Browser* tab, containing the transactions involving the selected file or directory.

## Browse Versions

Open a *Version Browser* tab, showing all the versions of the selected file or directory, and their interrelationships (ancestry).

## Send to Issue

Copies the selected entry(s) to another issue record's change package. If there's an existing entry for an element in a destination change package, an attempt is made to *combine* the existing entry with the entry you specified.

The *default query* is executed, and the results are displayed in a *dialog*. You are prompted to choose one or more of the issue records selected by the query. You can also create a new issue record, whose number will be entered in the dialog.

## Properties

Displays a message box with *information* about the selected element.


# The Issue Dependencies Tab

The Issue Dependencies tab contains a hierarchical display of the **dependencies** of a selected set of issue records -- that is, the set of issue records on which the selected records depend.

Note: in this description, "issue record N" is shorthand for "change package of issue record N".

From this tab, you can propagate issues' change packages to other streams, using simple *Promote* commands and/or the *Change Palette*. When propagating an issue's change package to another stream, you can choose to propagate the changes in the issue's dependencies, as well.

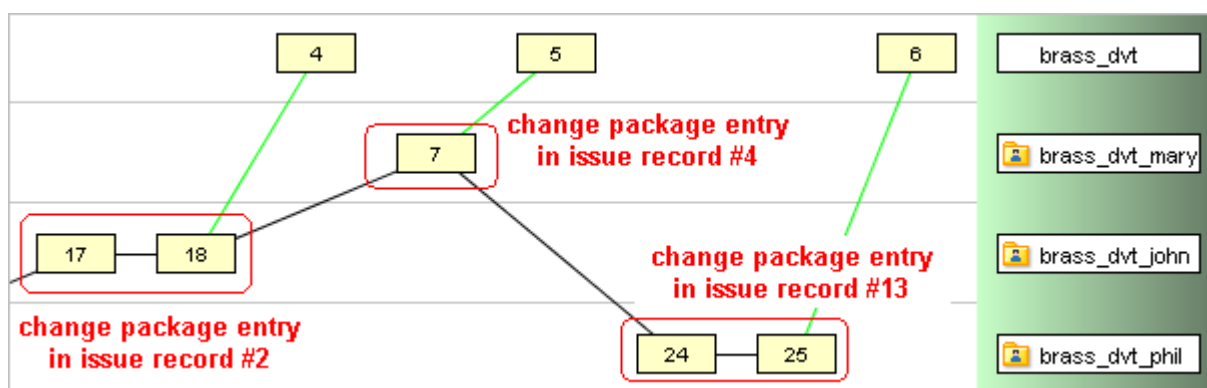
## Opening an Issue Dependencies Tab

In a *Stream Issues* or *Stream Diff (Issues)* tab, select one or more issue records in the Issues pane, then click the  *Show Dependencies* toolbar button or invoke the *Show Dependencies* command from the selection's context menu.

## Change Package Dependencies

The concept of a change package depending on another change package (or an issue record depending on another issue record) is complex, utilizing several aspects of AccuRev's product architecture. But the basic idea is familiar: when you ask for something, you sometimes get more than you asked for.

This Version Browser example shows three sets of changes to an element, each assigned to a different issue record: #2, #4, and #13.



At this point, issue record #13 is active in stream **brass\_dvt**. Promoting that issue record — for example, using the *Stream Issues* tab — promotes version **brass\_dvt/6** (real version: **brass\_dvt\_phil/25**).

Promoting issue record #13 would give you "more than you asked for": it would automatically promote issue records #2 and #4, as well. We describe this situation by saying that issue record #13 depends on issue records #2 and #4.

In practice, most issue records' change packages contain entries for multiple elements, not a single element. In many cases, the promotion causes issue records #2 and #4 to be "*partially in*" the destination stream, not "*completely in*".

## Dependency Defined

The discussion above is conceptual and informal. Here's a rigorous definition of change package dependency:

Issue A depends on issue B, in the context of a promotion from stream S1 to stream S2, if issue A is active in stream S1 and at least one element meets all the following conditions:

- The element has a change package entry in both issue A and issue B.
- The head version of the element's entry in issue A is a descendant of the head version in issue B. (That is, the A version contains the changes made in the B version.)
- Stream S2's version of the element is not either the head version of the element's entry in issue B or a descendant of it. (That is, the changes made in the B version have not yet been promoted to stream S2.)

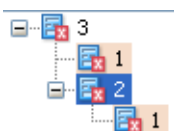
## The Dependency Context

Note that the definition of change package dependency cites a pair of streams as its context. The Issue Dependencies tab shows dependencies in the context of a promotion ...

- *from* the stream or workspace on which you opened a Stream Issues tab ...
- *to* its parent stream.

## The Dependency Hierarchy

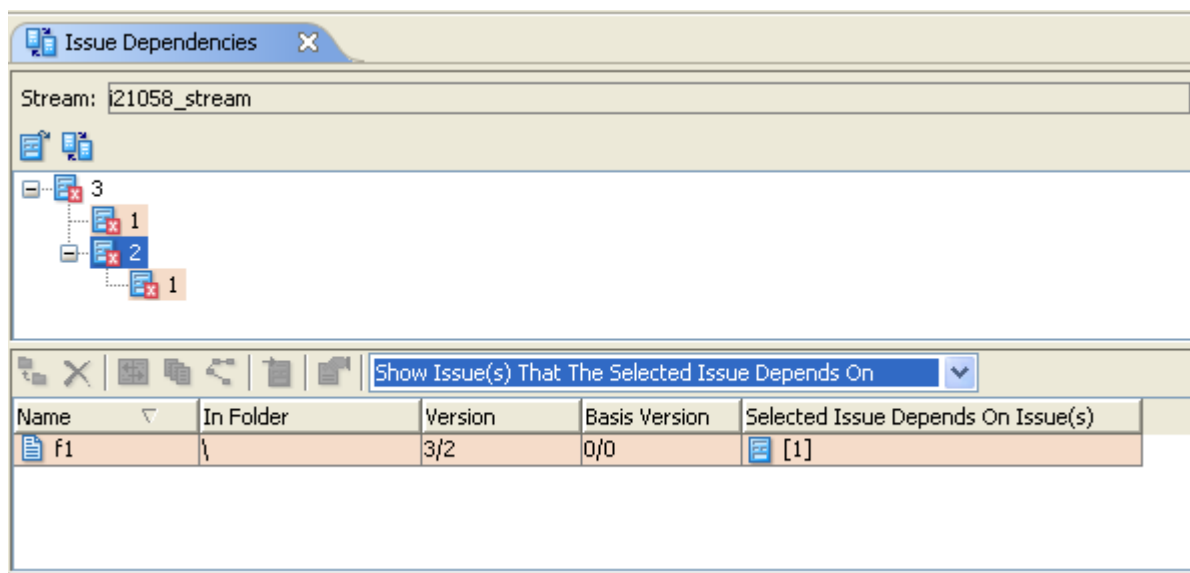
For each issue record, there is a dependency hierarchy: the issue directly depends on several issues; each of those issues directly depends on several issues; and so on.



The Issue Dependencies tab uses a tree control to represent the dependency hierarchy.

## Issue Dependencies Tab Layout

The Issue Dependencies tab layout is similar to that of the *Stream Issues tab*: two panes, each with its own toolbar. The upper pane is the Issues Pane, and the lower one is the Change Package Contents Pane:



## Issues Pane

The Issues pane displays the issue records that were selected for the *Show Dependencies* command.




Note: For each issue record, it displays the fields specified in the Change Package Results section of the Schema Editor's Change Packages tab.

If an issue record has dependencies, a tree control appears, initially in the collapsed state. Color-coding and icon differences indicate dependency-related information about the individual issue records:

As you expand issue records' dependency hierarchies, you may discover that some issue records that you initially selected for the *Show Dependencies* command also appear as dependencies at lower levels. To help you keep track, color-coding is used at the lower levels:

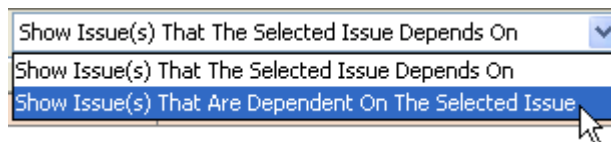
- Issue records that you initially selected for the *Show Dependencies* command appear without a tinted background.
- Issue records that you *did not* initially select appear with a tinted background. These are the "new records" added to the listing by the dependency analysis.

Variations in an issue record's icon provide more information about that issue's dependencies:

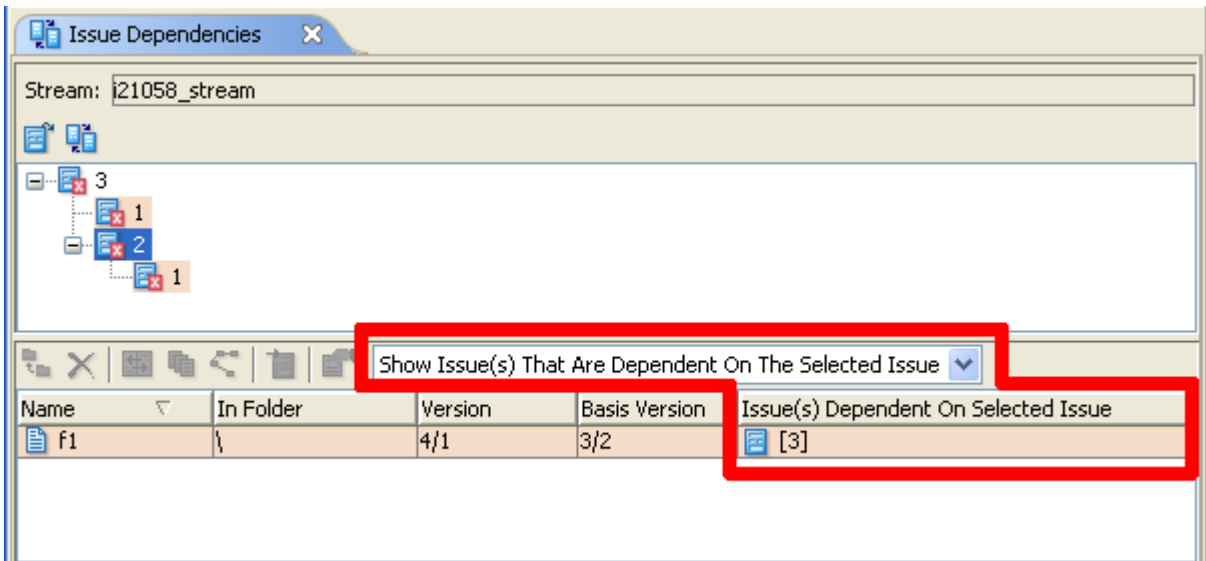
-  (no sub-icon): the issue has no dependencies in this context.
-  (yellow warning sub-icon): all of the issue's dependencies are included in the set you initially selected for the *Show Dependencies* command (the issues at the top hierarchy level).
-  (red warning sub-icon): some of the issue's dependencies are *not* included in the set you initially selected for the *Show Dependencies* command.

### Change Package Contents Pane

The information displayed in the Change Package Contents pane varies depending on the setting of the dependencies drop-down list.



Toogle between the two settings to display dependencies to and from the issue selected in the upper pane. This changes the information displayed in the bottom pane.



The far right column toggles to display the "Depends On" or "Are Dependent On" issues.

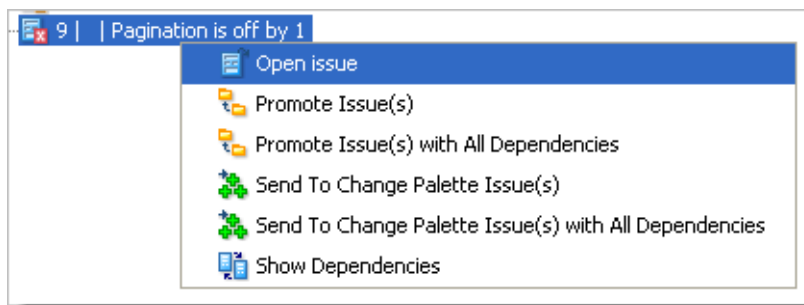
The versions are displayed as follows:

**Show Issue(s) That The Selected Issue Depends On** -- Displays the versions in those issues that the selected issue depends on.

**Show Issue(s) That Are Dependent On The Selected Issue** -- Displays the versions in the selected issue.

## Working in the Issues Pane

Use the tree control to navigate the dependency hierarchy of an issue record.



Select one or more issue records, and invoke any of these commands from the selection's context menu:

### Open Issue

Open an edit form on the selected issue record, and display the Changes tab.

### Promote Issue(s)

Promote each currently-active head version in the issue(s)' change package(s) to the parent stream. That is, if a version listed in the Version column is currently active in the stream

from which you invoked the *Show Dependencies* command, that version is promoted to the parent stream.

### **Promote Issue(s) with All Dependencies**

Similar to *Promote Issue(s)*, but also include the issue records on which the selected issue(s) depend.

### **Send to Change Palette Issue(s)**

Similar to *Promote Issue(s)*, but instead of promoting active versions, invoke the *Send to Change Palette* command on them.

### **Send to Change Palette Issue(s) with All Dependencies**

Similar to *Promote Issue(s) with All Dependencies*, but instead of promoting active versions, invoke the *Send to Change Palette* command on them.

### **Show Dependencies**

Open another Issue Dependencies tab, with the selected issue records at the top level.

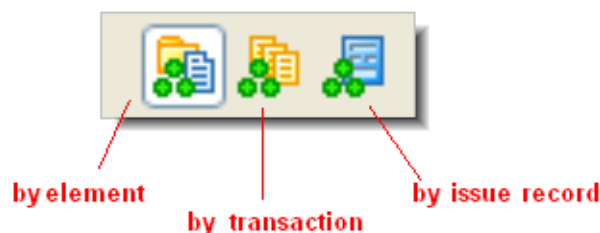
## **Working in the Change Package Contents Pane**

In this pane, you can perform the same operations as in the *corresponding pane* of a Stream Issues or Stream Diff (Issues) tab.

## **The Active Transactions Tab**

### **(Show Active Transactions command)**

When the *Stream Browser* is displaying element activity by transaction (not by element or by issue record), the *Show Active Transactions* command opens an Active Transactions tab, containing the transactions that created the versions that are currently *active* in a particular stream or workspace.



It may be the case that some, but not all, of the transaction's versions are currently active; the other versions may have already been promoted to the parent stream.

### **Notes:**

#### ***Multiple transactions for the same element***

If several *Promote* transactions created successive versions of the same element in a stream, and that element has not been promoted to its parent (or purged with *Revert to Backed*), then all of those *Promote* transactions will appear in the Active Transactions listing.

That is, a transaction is listed if the changes that it sent *to* the stream have not yet been promoted (or purged) out of the stream.


### ***"Checkout" and "Revert to Backed" transactions not included***

Transactions whose value in the Action column is **co** ("checkout") or **purge** are not included in the Active Transactions tab. Such transactions are created by the *Send to Workspace*, *Anchor*, and *Revert to Backed* commands.

### ***Comparison with Show History command***

The set of transactions displayed by Show Active Transactions is a subset of the transactions displayed by Show History. You can think of Show Active Transactions as displaying a stream's "current history", rather than its "ancient history".

## **Opening an Active Transactions Tab**

In the *StreamBrowser*, make sure development activity is being displayed by transaction (  button at right side of StreamBrowser toolbar). Choose *Show Active Transactions* from the context menu of a stream or workspace.

Note: You can't invoke this command on a depot's base stream. The "active" concept does not apply to the versions in this stream, because there is no higher-level stream to promote versions to, and to inherit versions from.]

## **Active Transactions Tab Layout**

The layout of the Active Transactions tab is virtually identical to that of the *History Browser*. Exception:

- The Versions pane includes an extra column, "Status". The **(member)** flag indicates that the element is currently active in the workspace or stream. The lack of this flag indicates that the element is no longer active (because it has been processed with the *Promote* command or the *Revert to Backed* command). This column also includes the **(overlap)** flag, if appropriate.

## **Working in a Active Transactions Tab**

The sets of commands available in the Summary pane (upper) and Versions pane (lower) of the Active Transactions tab are virtually identical to those available in the *History Browser*. The differences are:

### **Promote**

This command is enabled only in the Active Transactions tab, not in other History Browser contexts. In the Summary pane, it promotes the transaction's versions to the parent stream.

Note: Some of the transaction's versions may no longer be active, having been promoted or purged (*Revert to Backed* command). Invoking Promote operates on the versions that are still active.

In the Versions pane, it promotes the selected version (or set of versions) to the parent stream.


### **Revert**

The *Revert Transaction* command is enabled in other History Browser contexts, but not in the Active Transactions tab.

## The Stream Version Browser

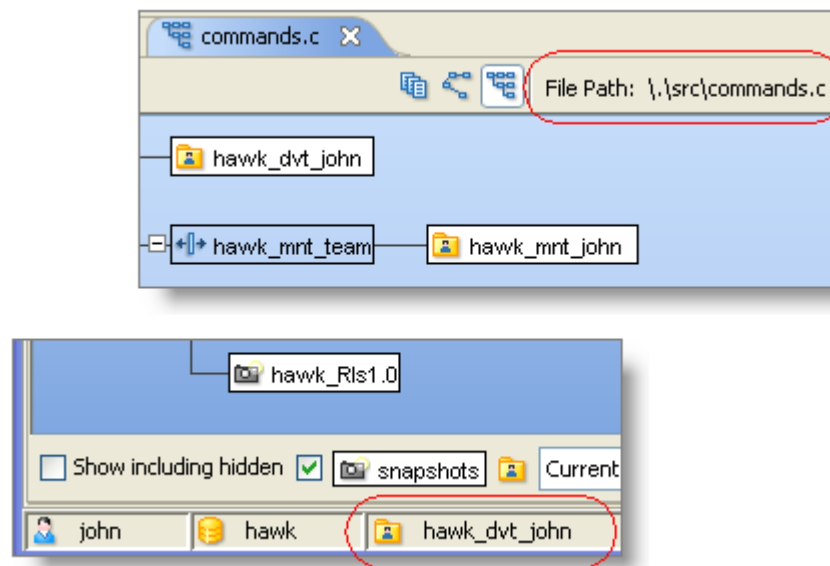
The Stream Version Browser tool is a variant of the *Stream Browser*. It displays the stream hierarchy for an entire depot; each stream, snapshot, and workspace in the display represents a version of a specified element.

### Opening a Stream Version Browser Tab

The Stream Version Browser displays information about one particular element. In a *File Browser*, select a file or directory element in the Details pane. Then click the  *Browse Stream Versions* toolbar button.

### Stream Version Browser Tab Layout

The Stream Version Browser display is almost exactly the same as the *StreamBrowser* display. But the pathname of the specified element is displayed in the upper right corner of the tab. Also, the workspace that you were in (when you invoked the Stream Version Browser from in the File Browser) is displayed at the bottom of the GUI window.



Each box in the Stream Version Browser display represents the version of the specified element that is currently in that stream, snapshot, or workspace. A workspace box indicates the version in the workspace stream, which is not necessarily the same as the file in the workspace tree. They differ if the file has **(modified)** status.

## Working in a Stream Version Browser Tab

The Stream Version Browser has many of the same commands as the *StreamBrowser* for controlling the display. These commands, available in the Stream Version Browser toolbar, include:

### **Search for Stream**

Find the stream(s) whose name includes a specified character string. A *Search for Stream* dialog appears.

### **Graphical Display**

Display the depot's stream hierarchy as a tree

### **Tabular Display**

Display the depot's stream hierarchy as a table

### **Graphical/Tabular Display**

Display the depot's stream hierarchy as a tree in one pane, and as a table in another pane.

Similarly, the Stream Version Browser includes many of the same commands as the *Version Browser*, for operating on a selected version of an element. These include:

### **Open**

Run the appropriate command on the version, according to its file type.

### **View**

Open a text editor on a temporary copy of the currently selected version (text files only).

### **Save As**

Copy the currently selected version to another filename.

### **Merge From**

Merge the selected version into the version in the workspace from which the Stream Version Browser was invoked.

### **Patch From**

Patch the selected version into the version in the workspace from which the Stream Version Browser was invoked.

### **Send To Workspace**

Activate the selected element in the workspace from which the Stream Version Browser was invoked.

### **Send To Issue**

Record the selected version in the *change package* section (Changes tab) of one or more *issue records*. The *default query* is executed, and you are prompted to choose one or more of the records selected by the query. You can also create a new issue record, to which the selected version(s) will be sent.

## **Show Difference**

Compare the selected version with another version of the element. AccuRev changes the mouse pointer, to prompt you to select the other version.

## **Properties**

Displays a message box containing information about the selected version.

# Patches and Change Packages

A patch is a set of versions of a text-file element, whose changes have been incorporated into another version of the element with the *Patch* (or *Patch From*) command. Informally, a patch consists of one user's "recent changes" to the element. The patch facility was designed to easily transfer to another development context the set of changes that were made to an element for a particular development task.

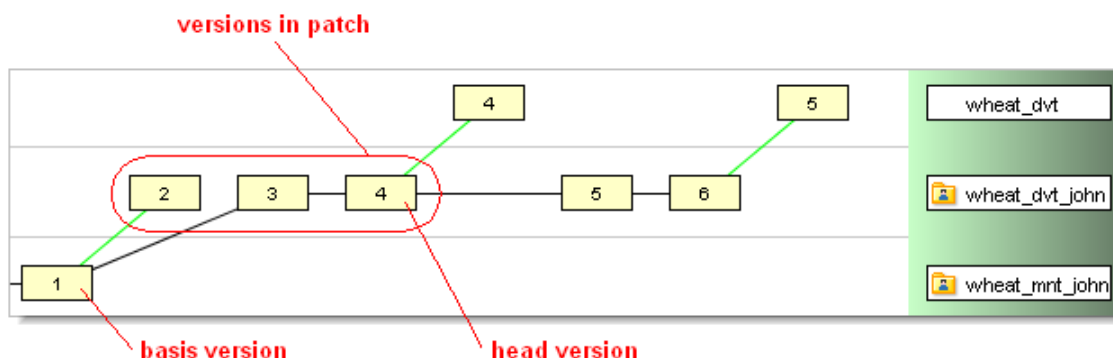
A change package contains a set of entries, each of which (usually) has the same form as a patch. But change package entries can be *modified* (added to) after they are first created. Such modified entries don't fit the "recent changes" description; rather, a change package entry can act like an accumulator, recording *all* the changes made to the element to fulfill a particular development task. The change package facility was designed to track exactly what element-by-element changes go into the bugfix or feature that the issue record describes.

## Structure of a Patch

The *Patch* command creates a new version in a workspace. This version is termed the **head version** of the patch. AccuRev automatically determines a corresponding **basis version** by scanning backward through the element's *ancestry*. The basis version is the most recent version that was either:

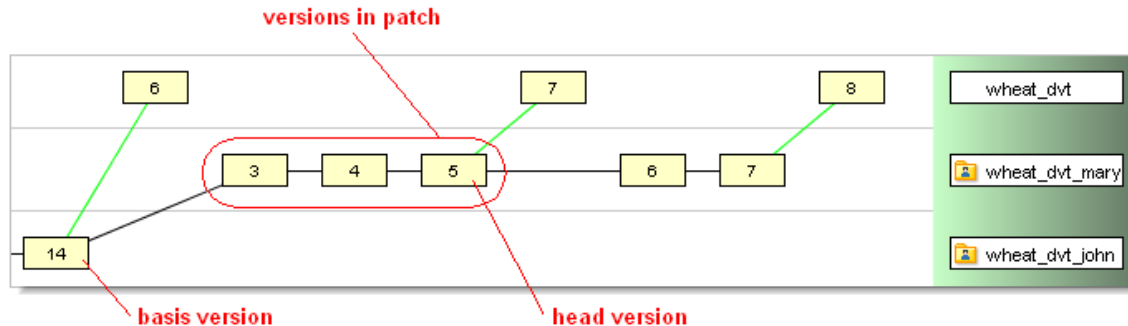
- originally created in another workspace
- created in the same workspace, then was promoted to the backing stream

**Example 1:** If patching from version *wheat\_dvt\_john/4*, AccuRev determines that John's "recent changes" to this element started with version *wheat\_mnt\_john/1*. (John used *Send to Workspace* to bring this "mnt" version into his "dvt" workspace.)

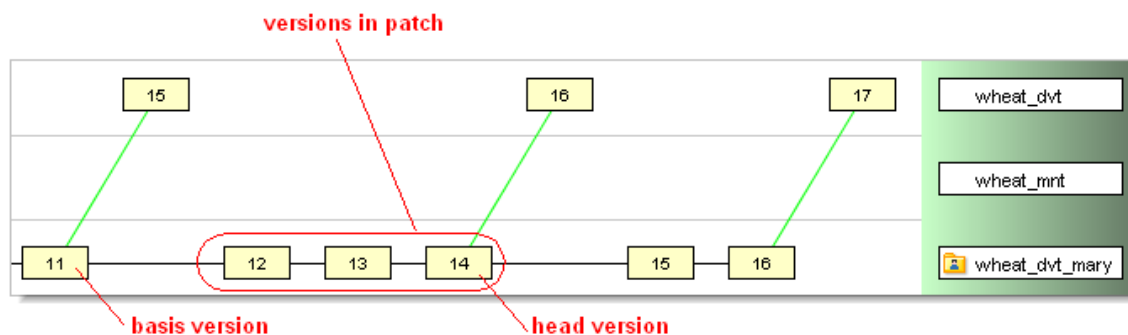


**Example 2:** If patching from version *wheat\_dvt\_mary/5*, AccuRev determines that Mary's "recent changes" to this element started with version *wheat\_dvt\_john/14*. (John created that version and

promoted it to backing stream *wheat\_dvt*. Mary brought the version into her workspace with an update.)



**Example 3:** If patching from version *wheat\_dvt\_mary/14*, AccuRev determines that Mary's "recent changes" to this element started with version *wheat\_dvt\_mary/11*. (Mary created that version -- maybe for the same development task, but maybe not -- and promoted it to backing stream *wheat\_dvt*. Then she continued working on the element.)



The patch consists of all the versions between the head version and the basis version. (The head version is included in the patch; the basis version isn't.)

### Notes:

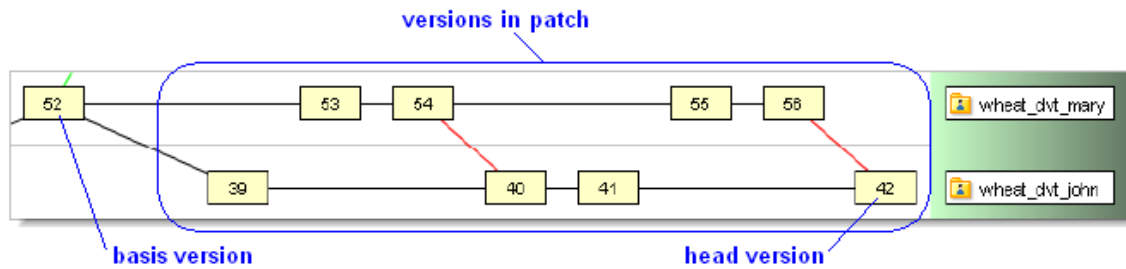
*Patches involve real versions, not virtual versions*

AccuRev's patch facility involves working with the contents of text files. And AccuRev records changes to files' contents as *real versions* in workspaces. Accordingly, AccuRev only uses real versions in its patch manipulations. If you specify a *virtual version* in some dynamic stream as the "patch from" version, AccuRev automatically uses the real version for which the virtual version is an alias.

*What it means to be "between" the head and basis versions*

In the simplest case, a patch's head version was derived from its basis version by a series of *Keep* commands (and maybe a *Rename*). In this case, it's easy to determine the set of versions that are between the head and basis versions. But what if one or more of the versions was created with *Merge*? Here's an example:

What if we "patch from" version *wheat\_dvt\_john/42*?



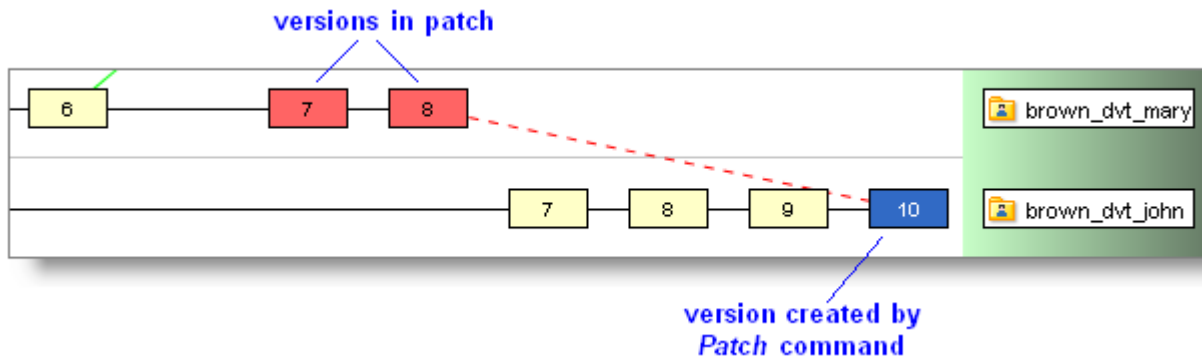
The basis version corresponding to this head version is the most recent one created in another workspace and brought into workspace *wheat\_dvt\_john* by an update -- that is, version *wheat\_dvt\_mary/52*. What versions are "between" these two versions? AccuRev determines the answer by finding this set difference:



In this case, the difference between the two subsets of the element's version graph consists of eight versions, from two different workspaces.

## Patch Display

In the *Version Browser*, a version created by *Patch* is connected to the "patch from" version with a **dashed red** line. Selecting this version highlights the versions in the patch.

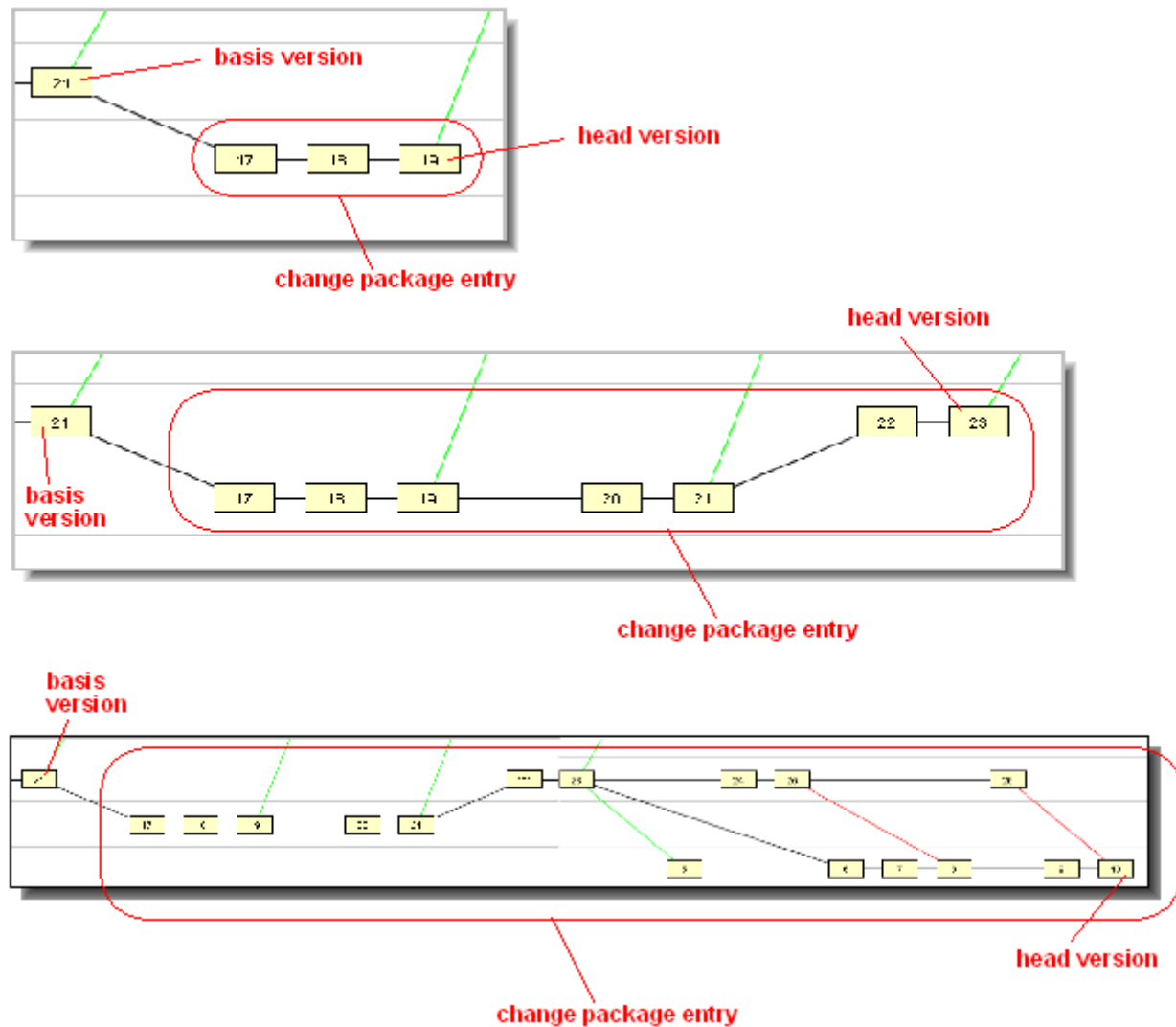


## Structure of a Change Package

Each *issue record* in an AccuWork *issue database* includes its own change package. A change package can contain entries for any number of elements -- but only one entry per element. The elements must belong to the same depot as the issue database.

Each change package entry is similar to a patch -- it consists of all the element's versions between a designated head version and basis version. This is a more general structure, because there is no "one user's recent changes" restriction to the scope of versions. The only restriction to is that the

basis version be an *ancestor* of the head version -- so that the set of "between" versions is well-defined.



## Change Package Display

In the *Changes subtab* of an issue record's *edit form* (and equivalently, in a *Stream Issues* tab), a change package is displayed in a table, each row showing one change package entry. Example:

Name	In Folder	Version	Basis Version
brass.c	\src\	5/13	5/10
chap03.doc	\doc\	5/11	4/7
tools.readme	\tools\	5/9	12/3

As with a patch, each change package entry has a head version (Version column) and a basis version.

## Creating a New Change Package Entry

Typically, a new change package entry is *exactly* like a patch -- the *Send to Issue* command and the *change package-level integration* between AccuRev and AccuWork create an entry whose basis version is automatically selected in the same way as the *Patch* command. When the entry is later modified, it assumes the more general structure.

You can create a new change package entry with the more general structure using the "*specifying basis*" variant of the *Send to Issue* command.

## Modifying Existing Change Package Entries

Change packages are designed to track ongoing changes to elements, not just a single set of changes. This means there will be times when you want to add a change package entry for a particular element, but an entry for that element already exists in the change package. In such situations, AccuWork attempts to combine the new entry with the existing one, producing an updated entry that includes all the changes. (Recall that there can be at most one entry for a given element in a given change package.)

### A Little Bit of Notation

To help explain how AccuWork performs "change package arithmetic" to combine and update entries, we'll use a simple notation. Suppose a change package entry contains the set of an element's versions defined by these specifications:

the head version is  $H$   
the basis version is  $B$

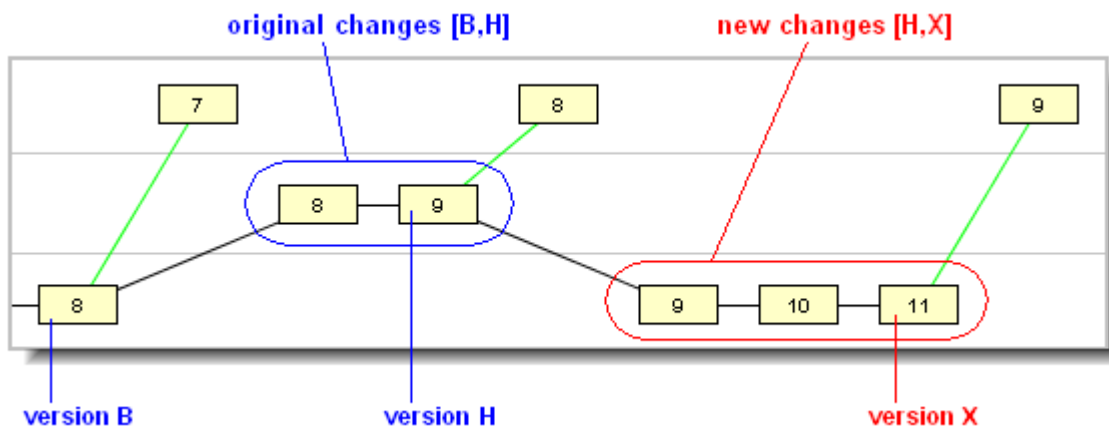
We'll use the ordered pair  $[B,H]$  to indicate this change package entry.

### Three Ways to Modify a Change Package Entry

Now, suppose a new change to an element is to be combined with the existing change package entry  $[B,H]$ . There are several cases, each handled differently by AccuWork:

#### Case 1: $[B,H] + [H,X]$ (simple extension of a change package entry)

This simple case typically arises when you think you're done with a task and record your work as change package entry  $[B,H]$  -- but it turns out that more work on the same element is required. So you (or a colleague) start where you left off, with version  $H$ , and make changes up to version  $X$ . Then, you want to incorporate the new set of changes  $[H,X]$  into the same change package.



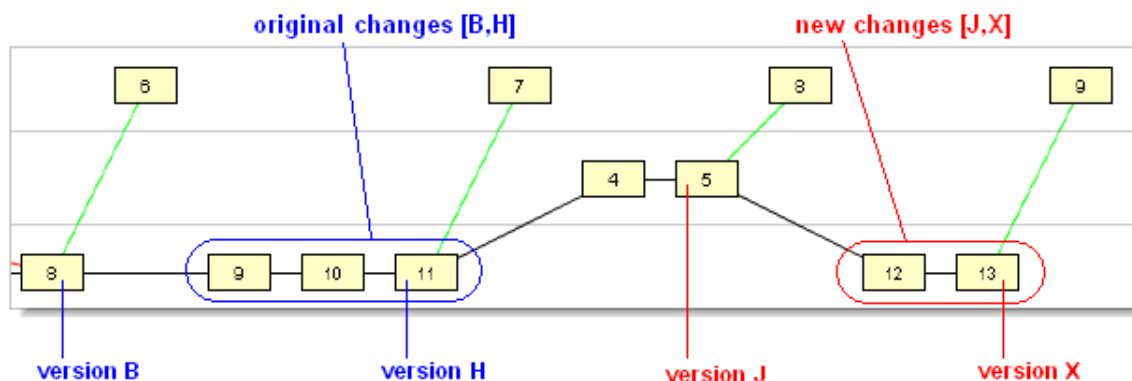
In this case, it's clear that the two series of changes can be viewed as a single, uninterrupted series -- starting at version  $B$  and ending with version  $X$ . That is:

$$[B,H] + [H,X] = [B,X]$$

Accordingly, AccuRev updates the change package entry automatically -- keeping  $B$  as the basis version and changing the head version from  $H$  to  $X$ .

#### Case 2: $[B,H] + [J,X]$ (where $H$ is an ancestor of $J$ : "change package gap")

This case typically arises when you do work on a task at two different times, and someone else has worked on the same element in between.

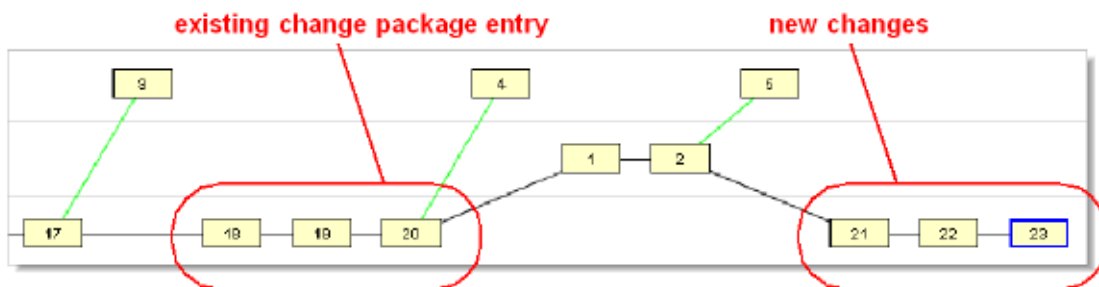


In this example, a colleague updated her workspace to bring in your original changes, created versions 9 and 10 in her workspace, and promoted her changes. You then updated your workspace to bring in her changes, and made a new set of changes.

When AccuRev tries to combine the change  $[B,H]$  and the change  $[J,X]$  into a single change package entry, it detects that version  $H$  and version  $J$  are not the same, but that  $H$  is a direct ancestor of  $J$ . Thus, there is a simple "gap" in the potential combined change package entry (in this example, consisting of your colleague's versions 9 and 10).

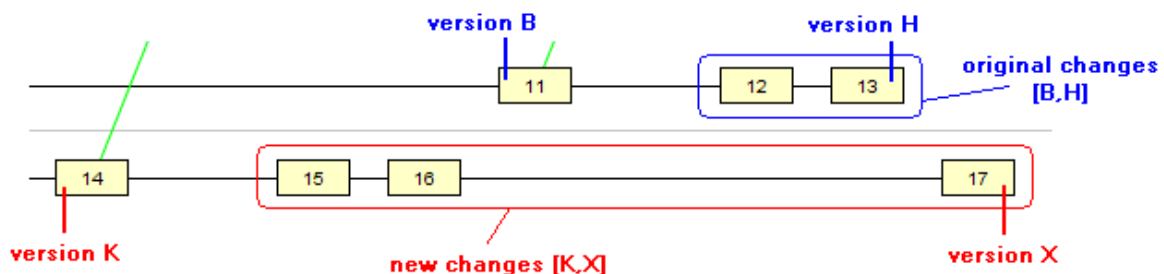
Probably, your colleague was not working on the same task when she made her changes. (If she had been, she would have added her changes to the same change package, as in Case 1.) On the other hand, it's probably OK to include the entire, uninterrupted series of versions  $[B,X]$  in your change set -- this includes both your original changes and your new changes (and, harmlessly, your colleague's changes, too).

Accordingly, AccuRev prompts you to approve this "spanning the gap" between the two change set entries, in order to create a single, combined entry.

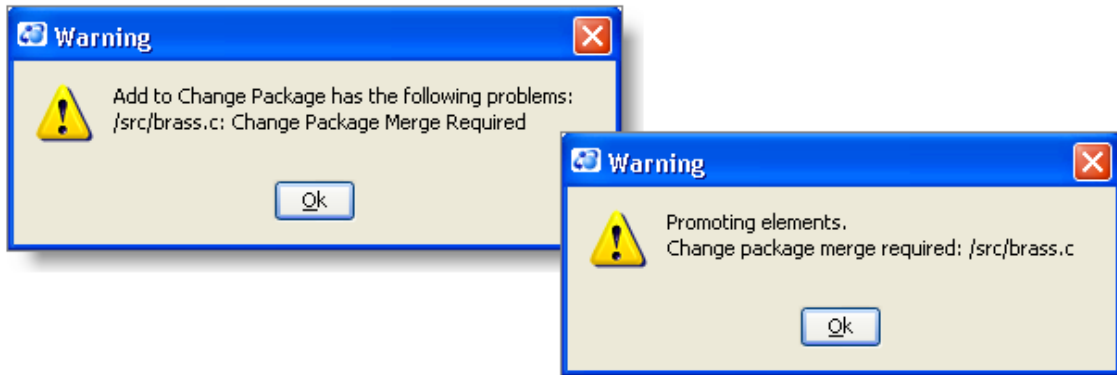


**Case 3:  $[B,H] + [K,X]$  (where  $H$  is not an ancestor of  $K$ : "change package merge required")**

This case typically arises when developers in workspaces that do not share the same backing stream try to use the same change package. There is no simple "gap" between the existing change package entry and the new one -- which means there is no way to combine them into a single change package entry (the basis version must be an ancestor of the head version).



AccuRev signals this situation with a "change package merge required" message, and cancels the current operation.



You can remedy this situation by performing a merge at the element level. (There is no merge operation defined at the change package level.) In the example above, merging version H and version X would create a new version; a change package entry with the new version as its head can be combined with the existing entry.

## Change Package History

How do you display the changes that have been made to a change package? Use the *Change Package History* icon on the Issues toolbar:



This brings up a list of all changes that have been made to the change package, including promotions, and elements that have been added ("cpkadd") or removed ("cpkremove").


Time	Action	User	#	Comment
2010/7/30 9:50:37	cpkadd	testuser1	18	
2010/7/30 9:48:28	cpkadd	testuser1	17	
2010/7/30 9:44:12	cpkremove	testuser1	16	
2010/7/29 12:13:26	promote	testuser1	13	

Name	In Folder	Head Version	Basis Version
f1	\		

For information about specific changes, select the transaction in the upper pane and view the elements associated with that change in the lower pane.

## Change package -- part of an issue record, or just associated with it?

An issue record's change package differs from its user-defined fields (Status, Severity, Description, etc.). When you change the value of a user-defined field, the  *Save* button is enabled in the edit-form toolbar. You can discard the change by closing the edit-form tab without

performing a *Save*. But the commands that create, modify, and delete change package entries take effect immediately. There is no way to discard such changes, and there is no need to *Save* them. (You can *Remove* a change package entry altogether, but you can't undo the adding of a new change to an existing entry.)

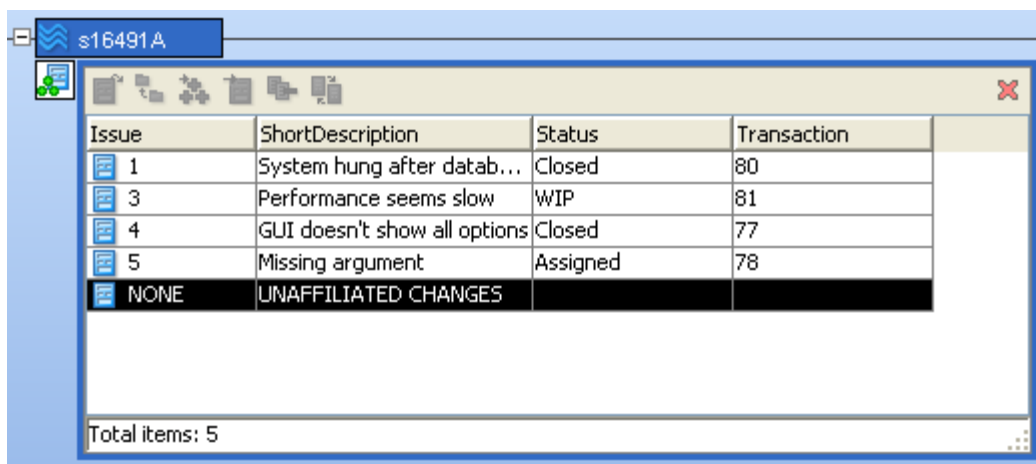
Change packages don't participate in the AccuWork's issue-history capability. Modifications to a change package don't appear in the *Issue History subtab* of an edit form (although you can use the *Change Package History* feature described above). And if you use this subtab to view an "old" version of an issue record, the Changes subtab still displays the current contents of the change package, not the "old" contents.

Given this behavior, you may want to think of a change package as being associated with an issue record, rather than being part of it.

## Unaffiliated Changes ("Dark Matter")

It is possible to end up with changes that are not associated with any issue. For example, somebody might decide that a file does not belong to an issue and remove it from the change package. Or somebody might decide to promote a file in a backing stream, without associating it with an issue. If such element versions are not associated with any issue, they are considered *unaffiliated* (sometimes called "dark matter"). Such element versions can lead to confusion, so AccuRev provides a way to view unaffiliated changes in the GUI.

In the stream browser, select Show Active Issue mode:



If any unaffiliated changes exist in the stream, the issue table gains an extra row labeled "NONE". If your table layout includes the ShortDescription column, you will also see "UNAFFILIATED CHANGES" in the row. Clicking this row displays an error message informing you that "This issue cannot be opened because it doesn't really exist".

There are two ways to deal with unaffiliated changes: short-term and long-term. If you are working under a deadline and just need a way to deal with these files so that they don't keep you from promoting everything in a stream, you can, for example, just **Promote by File** without associating the files with an issue. However, this is just a temporary fix, and you are just perpetuating the problem and pushing it further up your streams.

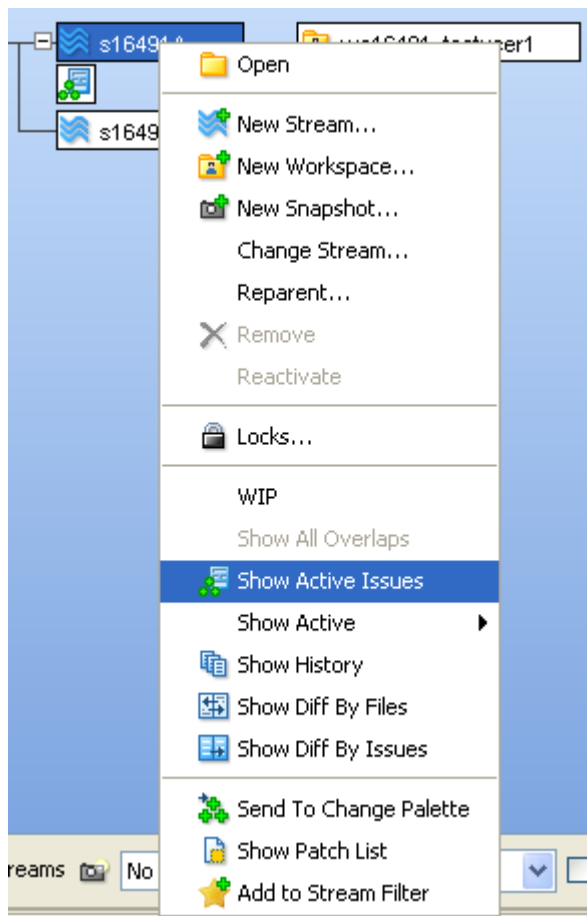
The long-term, correct solution is to determine why these changes are unaffiliated, and depending on how they got into this state, attempt to correct them. Typically there are three ways that an unaffiliated change may occur:

- somebody did a **Promote By File** without specifying an issue
- somebody removed a file version from a change package (either intentionally or accidentally)
- somebody patched a Change Package entry into a Workspace via the **Change Palette** (Path to a Tracking Issue)

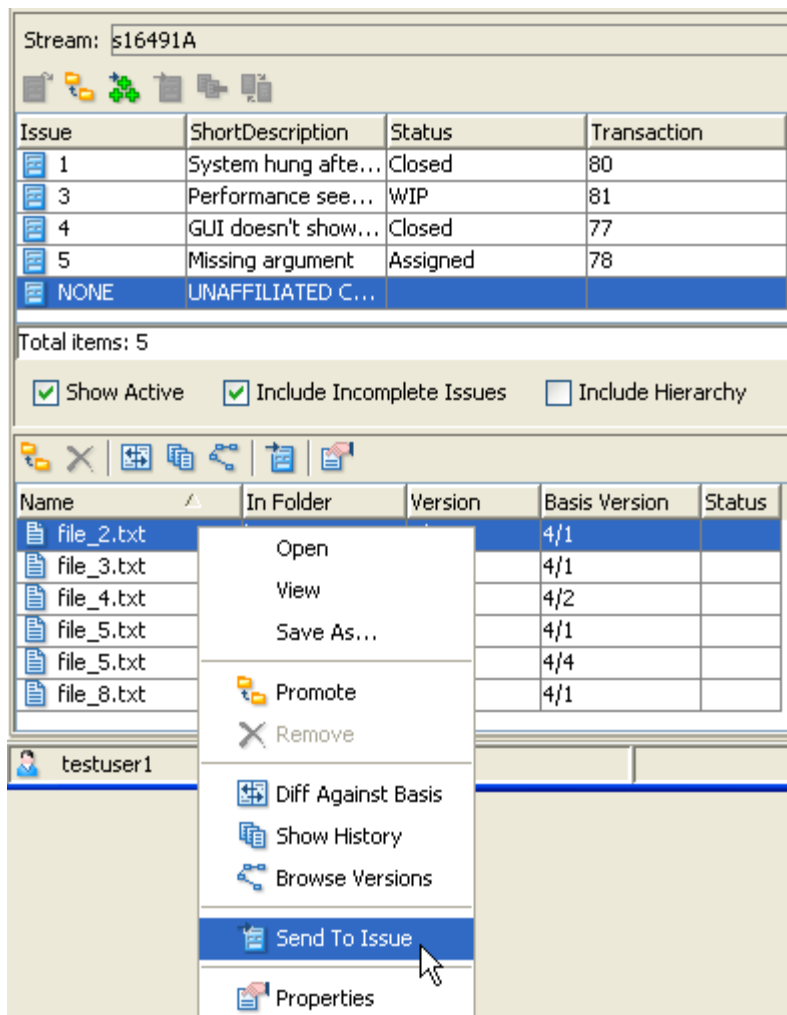
Once you have committed to working with change packages, you should never do **Promote By File** operations without associating the files with an issue. These can result not only in unaffiliated changes, but they can also cause incomplete change packages which are another source of confusion (see the "Incomplete Change Packages" chapter of the *AccuRev Technical Notes* manual for more information).

In any case, the best way to proceed is to associate the unaffiliated change with an issue, either by using **Send to Issue** for an existing issue, or by creating a new issue to handle this specific change.

To examine the unaffiliated versions in detail and deal with them, you should right-click the stream and click **Show Active Issues**.



From the resulting display, you can right-click the unaffiliated elements, explore their history, and then take appropriate actions to correct their unaffiliated status. For example, you can use **Send to Issue** and then promote, or promote the files to an issue while promoting from this stream to its parent.



## The Patch List Tab (Show Patch List command)



The Patch List tab lists all the individual versions whose changes need to be sent from one stream (or snapshot, or workspace) to another, so that the second stream will include all the changes in the first stream.

*Notes:*

- *Comparison with Stream Diff*—You can think of the Patch List tab as providing an "exploded view" of the Stream Diff (Files) tab. Whereas Stream Diff provides a simple "before" and "after" perspective on how each element changed, Patch List shows all the individual steps (that is, the versions) in progressing from the "before" state to the "after" state.

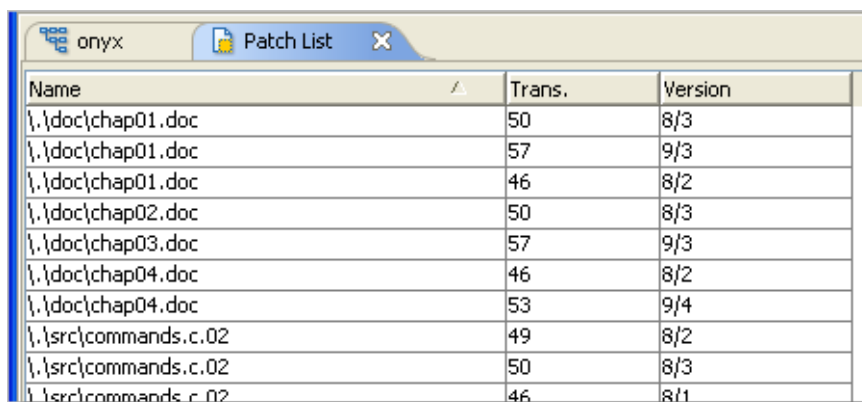
- *Patch List and the Patch command*—A Patch List table with 24 entries does not mean that you would need 24 invocations of the Patch command to propagate all the changes from one stream to the other. Each invocation of Patch incorporates all the changes in the specified patch A selected set changes (typically, the 'recent changes' made by one user) to a text-file element. Also, the merge-like operation that incorporates those changes in another version of the same element. See merge, basis version, head version, change package, reverse patch., which can include multiple versions. (Prior to AccuRev 3.5.5, each invocation of the Patch command did incorporate the changes in one version, so the correspondence to the Patch List table was exact.)

## Opening a Patch List Tab

1. In the *Stream Browser*, select a stream, snapshot, or workspace.
2. Choose the *Show Patch List* command from the selection's context menu, or click the  button on the StreamBrowser's toolbar.
3. The mouse pointer changes to . Left-click another stream, snapshot, or workspace.

## Patch List Tab Layout

The Patch List tab contains a table, each row of which details one version.



Name	Trans.	Version
\\.\doc\chap01.doc	50	8/3
\\.\doc\chap01.doc	57	9/3
\\.\doc\chap01.doc	46	8/2
\\.\doc\chap02.doc	50	8/3
\\.\doc\chap03.doc	57	9/3
\\.\doc\chap04.doc	46	8/2
\\.\doc\chap04.doc	53	9/4
\\.\src\commands.c.02	49	8/2
\\.\src\commands.c.02	50	8/3
\\.\src\commands.c.02	46	8/1

### Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

**Note:** This is the element's pathname in the first stream at the current time. This may not match its pathname at the time of the transaction listed in the Transaction column.

### Transaction

The *transaction* in which this version was created.

### Version

The *version-ID* of the version

## Working in a Patch List Tab

The Patch List tab display is informational only. See also: [Working with Tables](#) on page 9

## Streams and Issues

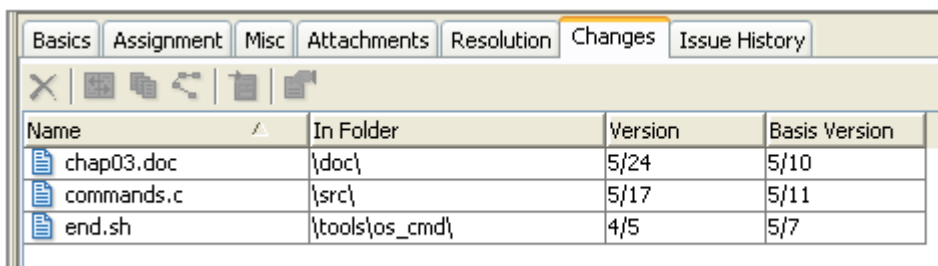
The *StreamBrowser* and *File Browser* enable you to see what individual versions are present -- and perhaps active -- in a stream. AccuRev also makes it easy to answer questions like these:

- "Are all the changes required to fix bug #4517 in this stream?"
- "Are any of the versions involved in the #4517 fix still active in this stream? (Or have all the changes already been promoted to a higher-level stream?)"
- "What new features are still under active development in this stream?"
- "The QA Group says they don't have all the Color Mixer changes for the upcoming release. Is that true?"

The key is to go beyond thinking of individual versions to considering collections of versions, called *change packages*. With AccuRev, change packages are implemented by AccuWork issue records. An issue record records the details of a bug or feature: its description, how important it is, who originated it, who's working on it, and so on. In the AccuRev Enterprise version of AccuWork, an issue record can also keep track of the changes that have been made to elements, in order to implement that particular bugfix or new feature.

For example, issue record #9 might contain a bug report, "Circles are not round". The bugfix involves changes to three elements:

When viewing the issue record through its edit form, go to the Changes tab to view the change package. In this example, the change package contains three entries:



Name	In Folder	Version	Basis Version
chap03.doc	\doc\	5/24	5/10
commands.c	\src\	5/17	5/11
end.sh	\tools\os_cmd\	4/5	5/7

**change package  
for issue record #9  
contains three  
changes**

- The set of changes that were made between version 5/10 and version 5/24 of element *chap03.doc*
- The set of changes that were made between version 5/11 and version 5/17 of element *commands.c*
- The set of changes that were made between version 5/7 and version 4/5 of element *end.sh*

See [Patches and Change Packages](#) on page 172 for the precise definition of "between".

### Change Packages "In" Streams

As the versions in a change package are promoted up the stream hierarchy, the change package itself implicitly moves up the hierarchy, also. Roughly speaking, a change package has risen to a certain level if all its entries have risen to that level. More precisely:

- The change package entry for an element is said to be "in" a particular stream if the element's version in the stream is -- or is a descendent of -- the head version (Version column). See also the definition of *direct/indirect*.

**Notes:**

*Why the "or is a descendent of" clause?*

AccuRev assumes that changes made to an element *remain* in that element as subsequent versions are created. Suppose a change package entry consists of versions 13/4, 13/5, and 13/6 of some element. Another user then brings version 13/6 into her workspace with an *Update*, and creates descendent version 49/2. It's fair to say that the change package entry is "in" her workspace, and in any stream to which version 49/2 is promoted.

*Why mention the head version, but not the basis version?*

The basis version of a change package entry is always an ancestor of the head version. AccuRev assumes that changes that were present in the ancestor version *remain* in the later version.

- A change package is said to be "completely in" a particular stream if all of its entries are "in".
- A change package is said to be "partially in" a particular stream if some -- but not all -- of its entries are "in". See also the definition of **incomplete**.
- A change package is said to be "active" in a particular stream if at least one of its head versions is in the stream's default group.

## 5. The History Browser



The History Browser displays some or all of the *transactions* associated with a particular *element*, *stream*, or *depot*. In addition to viewing information about a transaction and the *versions* involved in it, you can perform operations on the individual versions (*View*, *Diff*, etc.) and on the transaction's set of versions (*Promote*, *Revert*).


### Opening a History Browser Tab

There are several ways to launch the History Browser:

- To view the transactions involving an individual element

You can have the History Browser display the transactions in which one particular element was involved. (Other elements may have been involved in these transactions, too.) Elements are listed in various places in the GUI:

- In the Details pane of a File Browser tab
- In the Default Group Contents subwindow of a StreamBrowser
- In an issue record's change package (Changes subtab)

In any of these contexts, you can select an element and click the  *Show History* toolbar button. Alternatively, right-click an element and select *History > Show History* from the context menu.

- To view the transactions involving a particular stream

You can have the History Browser display transactions that affected a particular stream. For a *workspace stream*, this principally includes the *keep* transactions that create *real versions* in the stream. It can also include *co*, *move*, *defunct*, and *undefunct* transactions. For a *dynamic stream*, this includes *promote* transactions *to* the stream, but not promotions *from* the stream.

In the StreamBrowser, right-click a stream and select *Show History* from the context menu.

- To view the active transactions for a particular stream


Every stream has a *default group*, consisting of the elements that are active in that stream. In the Stream Browser, a stream's context menu includes the command *Show Active Transactions*; it opens a History Browser tab and loads the set of transactions (*Keep*, *Promote*, etc.) in which the stream's active versions were created.

- To view the transactions involving a particular depot

You can view the transactions for an entire depot -- all the elements, in all the streams. Use the *Admin > Depots* command to list all of the repository's depots. Then, select a depot and choose *Show History* from its context menu, or from the *Actions* group in the main menu.

- To view a single transaction or a specified group of transactions

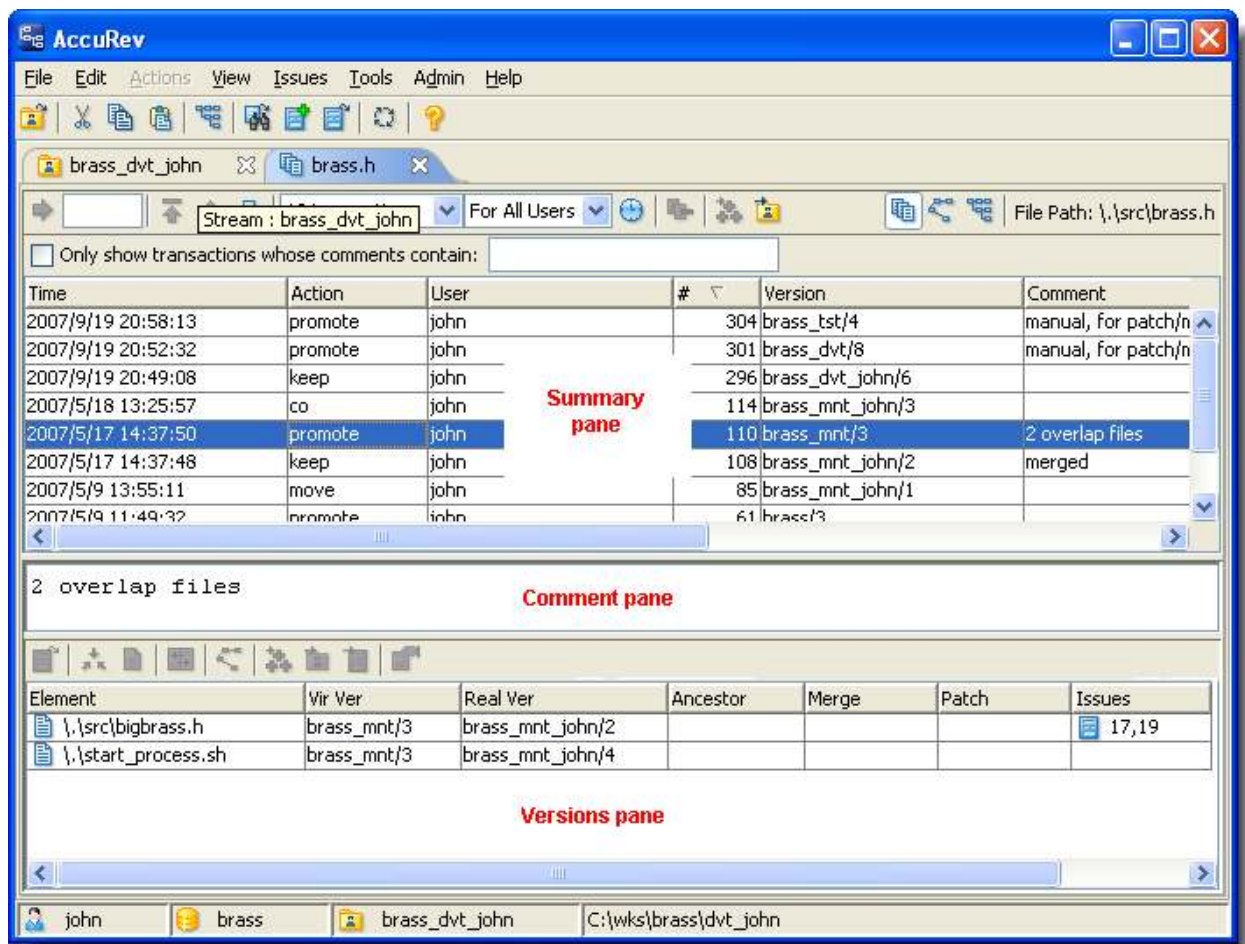
In the Version Browser, selecting *Show History* from any version's context menu displays the transaction in which that particular version was created.

If you have used AccuWork to promote files (see [Transaction-Level Integration between AccuRev and AccuWork](#) on page 331), an issue record's *affectedFiles* field contains a list of transaction numbers. Click the  *Show History* button next to this field to display just those listed transactions.

## History Browser Tab Layout

The History Browser appears in a separate tab of the AccuRev GUI's multiple-tab display. The tab is divided into three panes:

- The Summary Pane displays a group of transactions, one per line. This pane displays overall information: transaction number, timestamp, transaction type, etc.
- The Comment Pane shows the comment string, if any, that was specified for the currently-selected transaction.
- The Versions Pane shows all the versions that were involved in the currently-selected transaction. It also indicates which **change packages**, if any, those versions belong to (Issues column).



## The Summary Pane

The summary pane displays a table containing a set of transactions, one per row. Each row of the summary pane's table displays the following information about an individual transaction:

#### Time

A timestamp, indicating when the transaction took place.

#### Action

The kind of transaction: *keep*, *promote*, etc.

#### User

The *principal-name* of the user who initiated the transaction.

#### # (transaction number)

The unique number (within this depot) of the transaction.

#### Version

(only for transactions involving an individual element) The real version or virtual version of the element that was created in this transaction. This column doesn't appear in displays of a stream's history or an entire depot's history; the version(s) created by the transaction appear in the Versions pane.

#### Comment

The first line of the user-supplied comment for this transaction. If the comment extends to additional lines, an ellipsis ("dot dot dot") appears here. For the full text of the comment, look in the Comment pane.

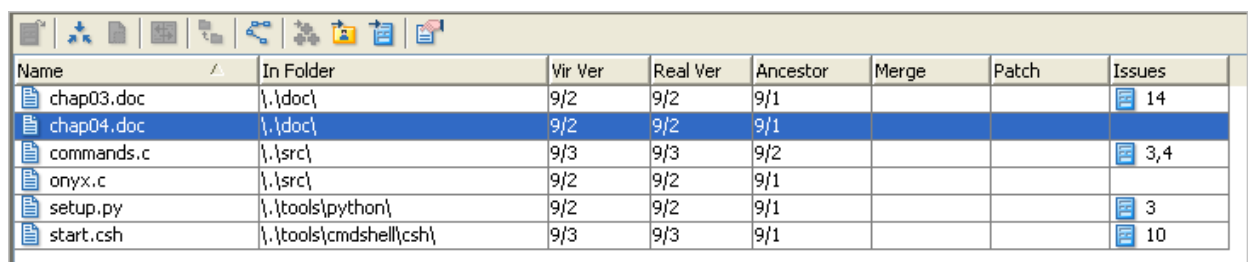
## The Comment Pane

The Comment pane displays the full text of the user-supplied comment for the transaction currently selected in the Summary pane.

Since transactions are immutable, you cannot change an existing comment in any way.

## The Versions Pane

The Versions pane displays a table of all the versions, if any, created by the currently selected (highlighted) transaction.



The screenshot shows a window titled 'Versions' with a toolbar at the top. Below the toolbar is a table with the following columns: Name, In Folder, Vir Ver, Real Ver, Ancestor, Merge, Patch, and Issues. The table contains six rows of data, with the second row highlighted in blue.

Name	In Folder	Vir Ver	Real Ver	Ancestor	Merge	Patch	Issues
chap03.doc	\\.doc\	9/2	9/2	9/1			14
chap04.doc	\\.doc\	9/2	9/2	9/1			
commands.c	\\.src\	9/3	9/3	9/2			3,4
onyx.c	\\.src\	9/2	9/2	9/1			
setup.py	\\.tools\python\	9/2	9/2	9/1			3
start.csh	\\.tools\cmdshell\csh\	9/3	9/3	9/1			10

#### Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

#### Virtual Version

## Real Version

The two *version-IDs* of this version. The real version-ID indicates the *workspace stream* where the version was originally created. The virtual version-ID is different if this transaction records:

- a promotion of the version to a dynamic stream (Promote command), or
- a check-out of the version to a workspace (Send to Workspace or Anchor command)

See [Real Versions and Virtual Versions](#) on page 137.

## Ancestor

(transactions in a workspace stream only) The *predecessor* version, which this version was derived from.

## Merge

(**keep** transaction recorded by the *Merge* command only) The "from" version to the merge operation that created this version.

## Patch

(**keep** transaction recorded by the *Patch* command only) The "from" version to the patch operation that created this version.

## Issues

The AccuWork *issue records (change packages)* to which this version has been added. This field indicates that the version is *currently* in those issue records' change packages. It does not reflect change package membership at the time the version was created, or at any other time in the past.

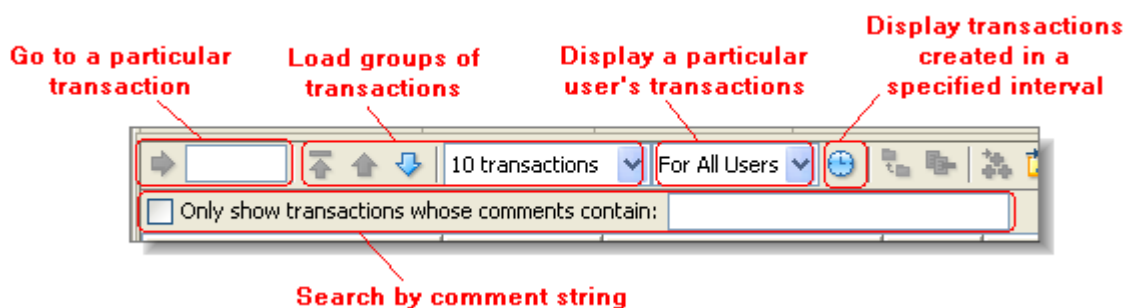
## Working in the Summary Pane

There's potentially a large number of transactions to display in the Summary pane -- for example, when displaying the history of an entire depot. Retrieving all the transactions at once from the depot's database can be time-consuming. The History Browser handles this situation by initially retrieving a small number of transactions, then letting you control the retrieval of additional transactions with the toolbar located just above this table.

Controls at the top of the Summary pane determine which transactions are displayed. You can invoke commands on one or more transactions that you select. See [Working with Tables](#) on page 9.

## Summary Pane Controls

The controls pictured below determine how many transactions (and which ones) are retrieved from the depot's database and loaded into the summary pane. Click on a control in the image below for more information about it:



Once a certain set of transactions has been loaded into the summary pane, you can browse through those transactions. Click any transaction to select it; the Comment and Versions panes are updated with the transaction's details. You can use the usual navigation keys to change the selected transaction: up-arrow and down-arrow, *PgUp* and *PgDn*, *Ctrl-PgUp* and *Ctrl-PgDn*.

### Summary Pane Commands

You can select one or more transactions in the Summary pane, and invoke commands on them. Use the selection's context menu or the toolbar buttons above the Summary pane.

Time	Action	User	#	Version
2007/9/19 20:58:13	promote	john	304	brass_tst/4
2007/9/19 20:52:32	promote	john	301	brass_dvt/8
2007/9/19 20:49:08	keep	john	296	brass_dvt_john
2007/5/18 13:25:57	co	john	114	brass_mnt_john
2007/5/17 14:37:50	prom	john		_mnt/3
2007/5/17 14:37:48	keep	john		_mnt_john
2007/5/9 13:55:11	mov	john		_mnt_john
2007/5/9 11:49:32	prom	john	61	brass/3

#### Revert

(from dynamic stream history only) For the selected *Promote* transaction, performs the equivalent of an "undo" in a particular child workspace of the dynamic stream. AccuRev prompts you to specify which workspace. Older versions of the transaction's elements will be activated in that workspace. For details, see the *revert* reference page in the *AccuRev CLI User's Guide*.

#### Diff Against Other Version

(from text-file element history only) Compare the version in the selected transaction with the version in another transaction. AccuRev prompts you to select another transaction from the summary pane.

#### Diff Against File in the Workspace

(from text-file element history only, when in the context of a particular workspace) Compare the version in the selected transaction with the file in the workspace from which the History Browser was invoked. This workspace is listed at the bottom on the AccuRev

GUI window. This enables you to invoke a comparison with a file that you have changed in your workspace, but have not yet preserved in the repository with a *Keep* command.

This command is disabled if you invoked the History Browser from a non-workspace context.

### **Send to Change Palette**

(*dynamic stream only*) Load the selected versions into the Change Palette, so that they can be merged and promoted to other streams.

### **Send to Change Palette (Changes)**

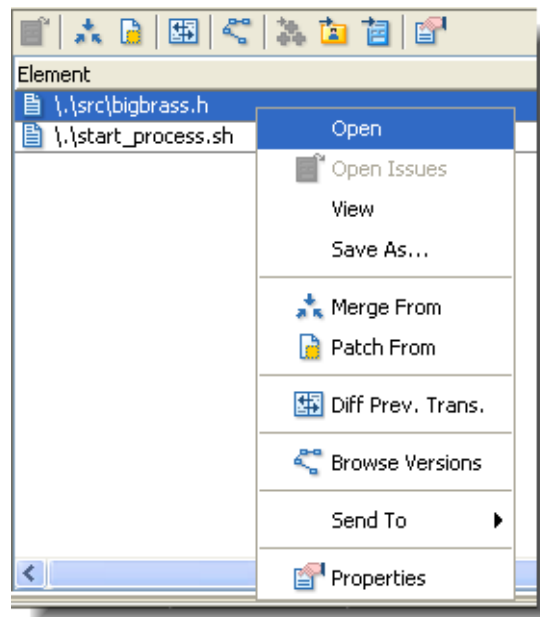
(*dynamic stream only*) Load the selected versions into the Change Palette, so that they can be patched to other streams.

### **Send to Workspace**

Activate each of the transaction's versions in a particular workspace. AccuRev prompts you to specify one of your workspaces.

## **Working in the Versions Pane**

You can perform several operations on a selected version in the Versions pane, using its context menu or the versions pane toolbar.



### **Open (equivalent to double-click)**

**Windows:** Run the appropriate command on the file, according to its file type. (The Windows file-typing system — "file associations" — does not provide for assigning a file type if the filename has no suffix.)

**UNIX:** Open a text editor on the file.

### **Open Issues**

Displays the change package(s) -- that is, the issue record(s) -- to which the version belongs.

Note: A version is "in" a change package if it -- or any of its descendants -- is the head version of an entry in that change package.

For a version that belongs to a single change package, an edit form opens on that one issue record. For a version that belongs to multiple change packages, an Issues tab opens, listing all the issue records.

## View

Open a text editor on a temporary copy of the currently selected file (text files only).

## Save As

Copy the currently selected file to another filename.



## Merge From

(only when displaying the history of an individual file) Merge the selected version into the version in the workspace from which the History Browser was invoked.



## Patch From

(only when displaying the history of an individual file) Patch the selected version into the version in the workspace from which the History Browser was invoked.



## Diff Against Previous Transaction

Compare the selected version with the version that was in its workspace or stream just before the version was created.



## Promote

(only available in the *Active Transactions* tab, not in other History Browser contexts)



## Browse Versions

Open a Version Browser on the element whose version you selected.



## Send to Change Palette

(dynamic stream only): Load the selected versions into the Change Palette, so that they can be promoted to other streams.



## Send to Workspace

(only when displaying the history of an individual file) *Activate* the selected element in the workspace from which the History Browser was invoked.



## Send to Issue

Create a new entry (or modify an existing entry) in the *change package* section (Changes tab) of one or more *issue records*. The version you specify becomes the *head version* of the change package entry; AccuRev determines the corresponding *basis version* automatically.

The default query is executed, and the results are displayed in a dialog (see [The Send to Issue Command](#) on page 129). You are prompted to choose one or more of the issue records

selected by the query. You can also create a new issue record, whose number will be entered in the dialog. Note: If you have AccuWorkflow enabled, and use a New Issue button from within a dialog in the Java client, you must set any AccuWorkflow fields in those issues manually.

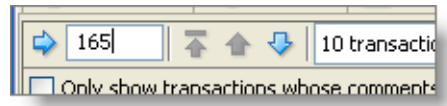
### **Properties**

Displays information about the selected element. The data items displayed vary with the type of element.

## History Browser / Summary Pane Controls

### Go to a Particular Transaction

Entering a transaction number in the Goto field unloads the current set of transactions, then loads the set beginning with the specified number.

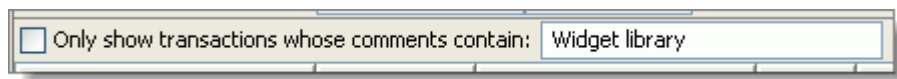


If that particular transaction is not relevant -- e.g. it did not involve the element whose history you're viewing -- the set begins with the next older relevant transaction.

## History Browser / Summary Pane Controls

### Display Transactions with a Particular Comment

By default, the Summary pane displays transactions no matter what comments they contain (if any). To filter the display based on transactions' comment fields, type a search string in the Comment Filter input field, then click the checkbox to enable the filtering.

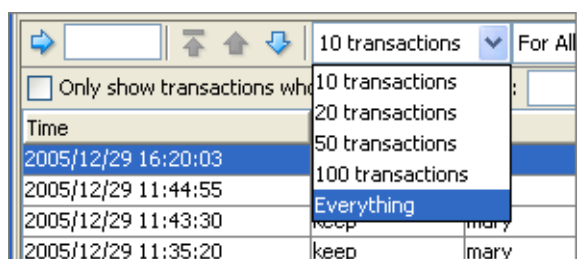


Note: either checking or unchecking the checkbox causes AccuRev to immediately repopulate the Summary pane.

## History Browser / Summary Pane Controls




### Manage Groups of Transactions

The Summary pane table initially contains the 10 most recent relevant transactions.




To change this count, use the Transaction Count list box. (Each time you change the count, the browser returns to displaying the most recent transactions.) Selecting *Everything* loads all the relevant transactions into the summary pane.

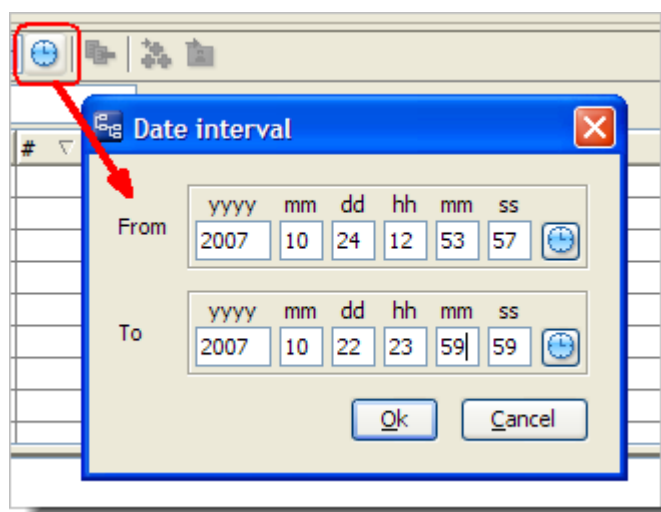
Several controls on the toolbar enable you to retrieve additional transactions, which have not yet been loaded into the summary pane:

- The  button unloads the current set of transactions, then loads the next older set of transactions.
- The  button unloads the current set of transactions, then loads the next newer set of transactions. This button is enabled only when you're viewing the history of an entire depot.
- The  button unloads the current set of transactions, then loads the first (newest) set of transactions.

## History Browser / Summary Pane Controls

### Display Transactions Created in a Particular Interval

By default, the Summary pane displays transactions no matter when they were created. Use the  *Set Date Interval* toolbar button to restrict the display to transactions that occurred in a specified interval.

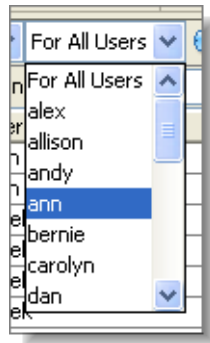


This changes the setting in the Transaction Count list box to *Custom* . To cancel the date interval restriction, change this list box setting back to *10 Transactions* (or something else).

## History Browser / Summary Pane Controls

### Display a Particular User's Transactions

By default, the Summary pane displays transactions created by all users.




To restrict the display to one user's transactions, use the User Filter list box. Select **For All Users** to turn off by-user filtering.

## The Version Browser

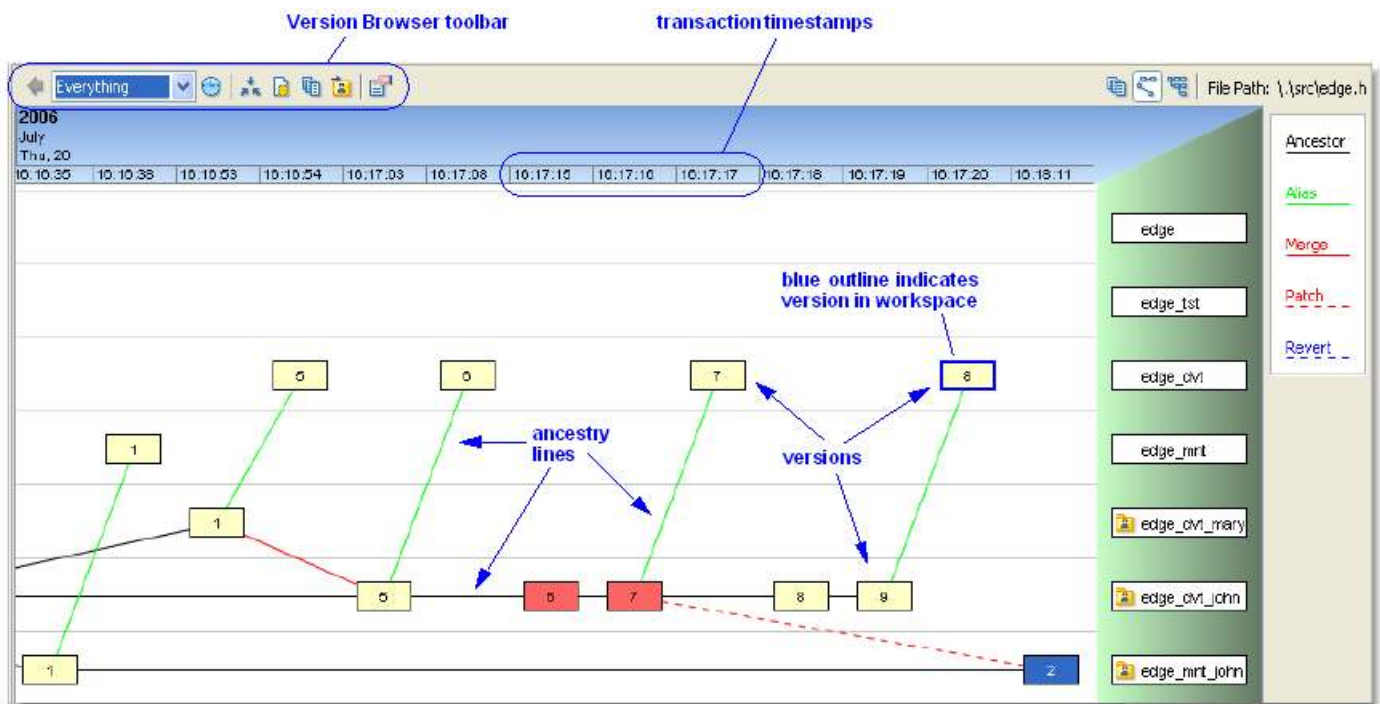
AccuRev maintains complete *ancestry* information for each *element*. The Version Browser displays some or all of an element's *versions*, using color-coded lines to indicate the way in which each version was created. You can perform various version-related operations, such as comparing any two versions (text files only) and copying any version to your workspace.

## Opening a Version Browser Tab

The names of elements appear in many contexts in the AccuRev GUI. These include the Details pane of the File Browser, the versions pane of the History Browser, and the Changes tab (change package) of a AccuWork issue record. In any of these contexts, you can select one element and invoke the  *Browse Versions* command. (In some cases, this command is a subchoice, under a *History* menu choice.)

# Version Browser Tab Layout

The Version Browser display contains:



- A yellow box for each version of the specified element. These boxes are arranged in rows, according to the workspace or stream in which the versions were created.

One yellow box has a blue outline. This is the version that currently appears in the workspace or stream from which you opened the Version Browser tab.

- A timestamp above each version box, indicating when the version was created. The tooltip for the version box itself contains the number of the transaction that created the version.
- The name of the workspace or stream, at the right edge of the display.
- Color-coded ancestry lines connecting the versions, indicating how the later version was derived from the earlier version.

## Ancestry Lines

The Version Browser uses these color-coded lines to indicate ancestry relationships

- *direct ancestor* (**black**) -- A user started with the earlier version, modified (or overwrote) it, and then performed a *Keep*, creating the later version. (There are a few other commands, including *Rename*, that might have created the later version. See below for details.)
- *alias* (**green**) -- A user promoted the earlier version to create the later version.
- *merge* (**red**) -- A user executed the *Merge* command to create the later version. There are always two earlier versions: the black line indicates the direct ancestor (see above) of the later version; the red line indicates the version that was merged with the direct ancestor.

- *patch* (**dashed red**) -- Similar to *merge* above; instead of the *Merge* command, the *Patch* command was used to create the later version.
- *revert* (**dashed blue**) -- The later version was created by a 'reverse patch', performed by the *Revert* command. This effectively 'undoes' the changes in a specified *transaction* or *change package*.

## Ancestry Relationships

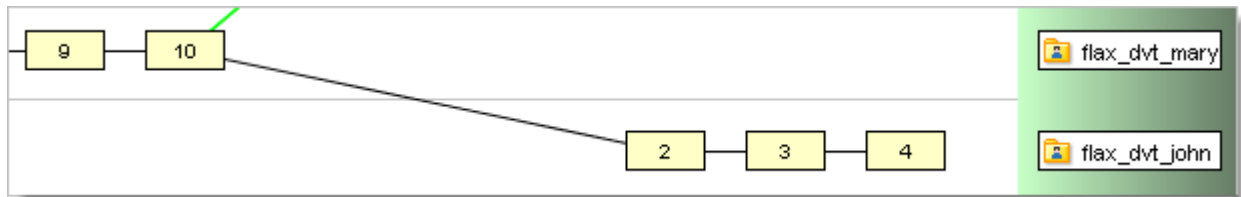
The following sections discuss the kinds of ancestry in more detail.

### Direct Ancestor -- Modification of an Existing Version

Probably the most common AccuRev operation is starting with an existing version, making changes, and then executing the *Keep* command to save the changes in a new version. The existing version might have been created by you -- for example, with a previous *Keep* command. Or it might have been created by someone else: that user *Promote*'d the version, then you brought it into your workspace with an *Update*.

The version in your workspace created by the *Keep* command is termed a *real version*, because it represents a change to the element. Other commands can create real versions in your workspace, too: *Rename*, *Defunct*, *Undefunct*.

The Version Browser uses a **solid black line** to connect the existing version (termed the direct *ancestor* or predecessor) with the new version.



In the figure above:

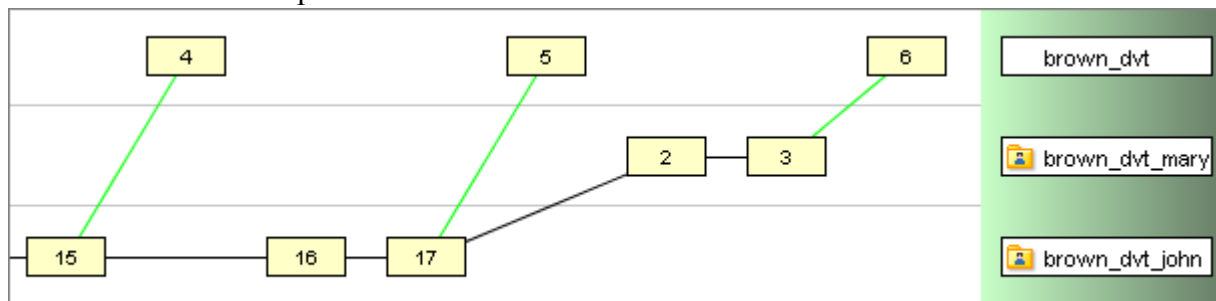
- Version 9 in the *flax\_dvt\_mary* workspace stream was revised to create version 10 in the same stream.
- Workspace stream *flax\_dvt\_john* was updated to incorporate version *flax\_dvt\_mary/10*. Then, this version was revised to create version *flax\_dvt\_john/2*.
- Version 2 in the *flax\_dvt\_john* workspace stream was revised to create version 3 in the same stream; and version 3 was revised to create version 4.

Exception: a version created by a *revert operation* is connected to its direct ancestor by a **dashed blue line**.

### Alias -- Virtual Version Ancestry

Workspaces contain *real versions*, which represent changes to elements. By contrast, all versions in dynamic streams are *virtual versions*, created with the *Promote* command. Each virtual version is an alias for -- that is, a pointer to -- some real version in a user's workspace. The Version

Browser uses a green line to connect a virtual version in a dynamic stream to the corresponding real version in a workspace.

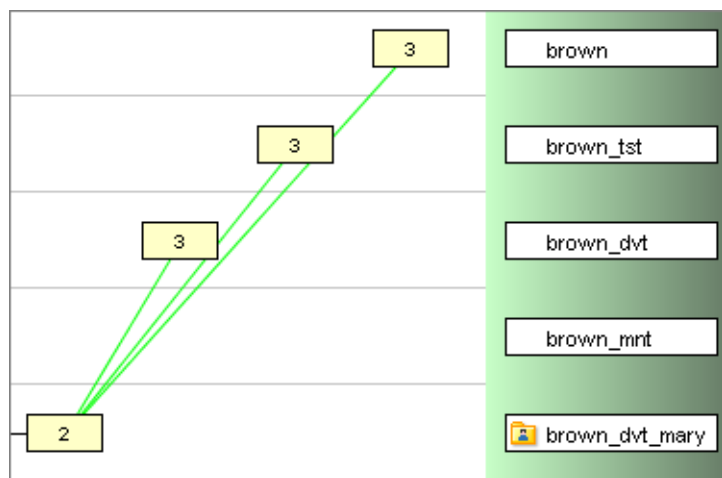


In the figure above:

- version 4 in stream *brown\_dvt* is an alias for (was promoted from) version 15 in workspace *brown\_dvt\_john*
- version 5 in stream *brown\_dvt* is an alias for version 17 in workspace *brown\_dvt\_john*
- version 6 in stream *brown\_dvt* is an alias for version 3 in workspace *brown\_dvt\_mary*

There's one exception to this scheme. The *Anchor* or *Send to Workspace* command creates a virtual version in a workspace. It's a virtual version because it doesn't represent a change to the element, but merely the restoration of an existing version to the workspace.

**Successive Promotions.** In a depot with a deep stream hierarchy, it's common to successively promote a particular version to the parent stream, then to the grandparent stream, then to the great-grandparent stream, etc.



All the versions created by this series of *Promote*'s are aliases for the same real version. The Version Browser shows how all the virtual versions relate back to the original real version.

The versions in streams *brown\_dvt*, *brown\_tst*, and *brown* are all aliases for the real version 2 in workspace stream *brown\_dvt\_mary*. (The display does not indicate the fact that the version was promoted from *brown\_dvt* to *brown\_tst*, and from *brown\_tst* to *brown*.)

## Merge -- Merging of Two Versions

A standard *merge* operation combines the contents of these two versions of a file:

- The most recently kept version in your workspace stream. This version contains the changes that you have made to the file in your workspace.

Note: It's possible -- but not a best practice -- to perform a merge on a file with (modified) status. Since you haven't preserved the recent changes with *Keep*, AccuRev will have no way to restore the file to its state just before the merge. This can be painful if you change your mind or make an error during the merge process.]

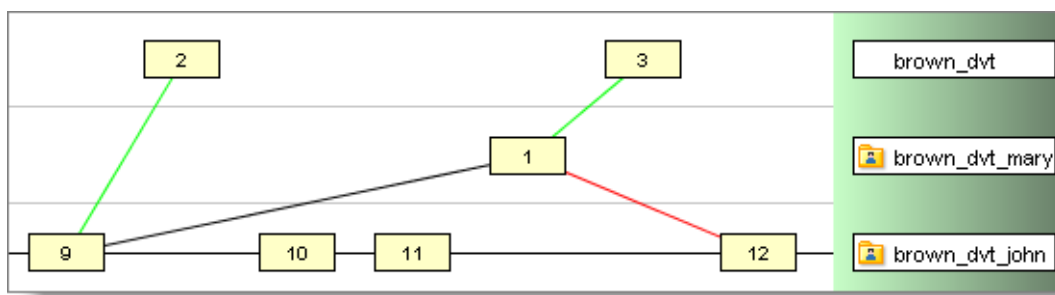
- The most recent version in the backing stream.

The result file of the merge operation is kept as a new version in the workspace stream. (You can think of merging as a fancy text-editing operation; as with any edit to a file, you preserve the results with *Keep*.) This new, merged version has two ancestors: the two versions listed above.

This is all simple enough. There's a twist, though, which shows up in the Version Browser display: AccuRev always records real versions, not virtual versions, as the two ancestors of a new, merged version. Thus, the ancestors in the standard merge scenario described above are:

- The most recently kept version in your workspace stream.
- The version in some other workspace stream that was promoted to the backing stream, causing the *overlap* that necessitated the merge.

**Example:** This screen shot shows a merge from the backing stream *brown\_dvt* to the workspace stream *brown\_dvt\_john*.



The new, merged version is *brown\_dvt\_john/12*.

Its ancestors are:

- Real version *brown\_dvt\_john/11*
- Real version *brown\_dvt\_mary/1*, which was promoted to become virtual version *brown\_dvt/3* in the backing stream.

A solid red line shows the merging of data from one workspace, *brown\_dvt\_mary*, to a different workspace, *brown\_dvt\_john*. The black line ("direct ancestor") between versions 11 and 12 in the *brown\_dvt\_john* workspace reflects the viewpoint that merging is just a fancy text-editing operation, automating the creation of the next version of a file in a workspace.

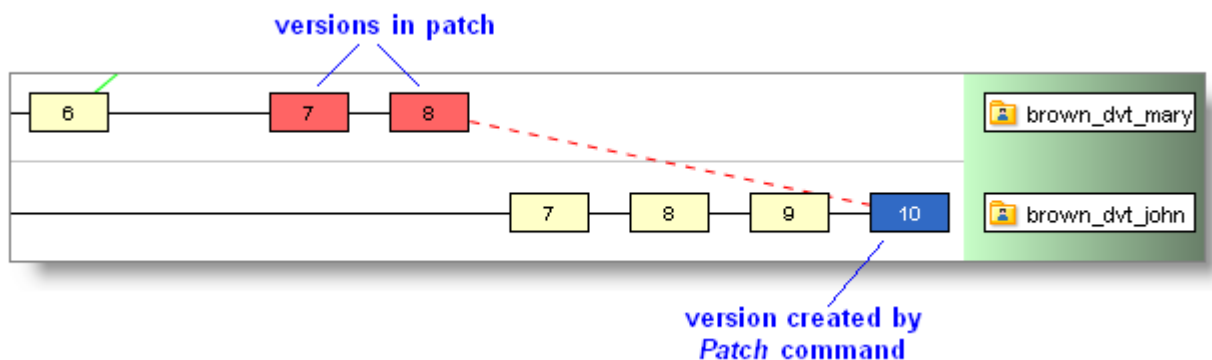
## Patch -- Selective Inclusion of Another Version's Changes

A *patch* operation is similar to a *merge* operation. In both, text from another version (the "from" version, at the left end of the **dashed red line**) is incorporated into your workspace's version.

Here's the difference:

- A merge operation considers the entire contents of the "from" version.
- A patch operation considers only *some* of the changes in the "from" version. Typically, it's the 'recent changes' made by one user, recorded in one version or a set of consecutive versions. See [Patches and Change Packages](#) on page 172 for full a discussion of patches.

When you select a version created by the *Patch* command, the Version Browser highlights in red the versions contained in that patch.



*Note: Patching and the closest common ancestor*

AccuRev tracks patch ancestry separately from merge ancestry. In determining the *closest common ancestor* of two versions for a *merge* operation, AccuRev takes into account previous merge operations (solid red), but not previous patch operations (dashed red) or revert operations (dashed blue).

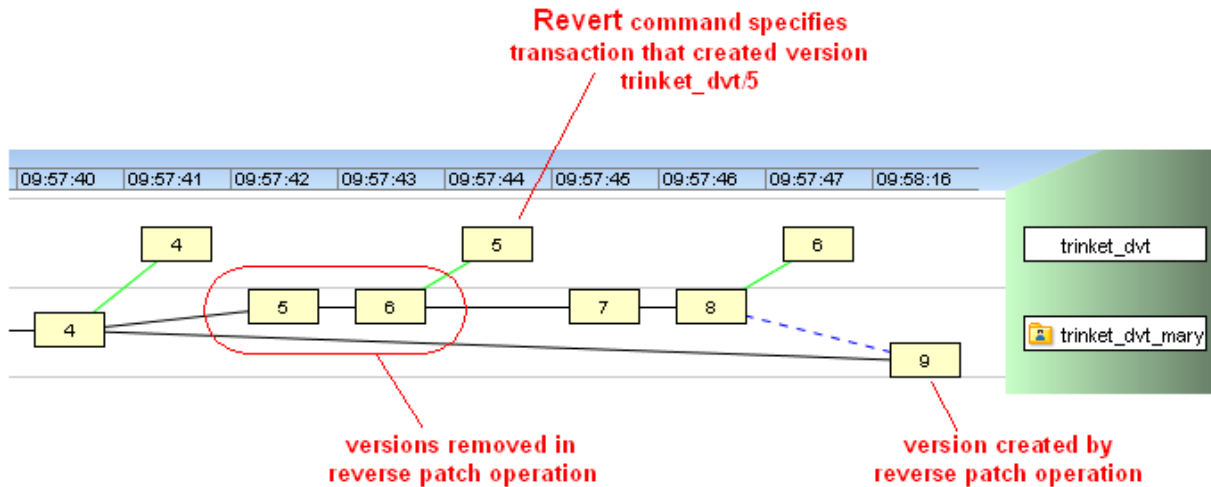
## Revert -- Selective Removal of Changes from a Version

A *revert* operation is the opposite of a *patch* operation. (And we describe the *Revert* command as performing a 'reverse patch' operation.) Whereas a patch *adds* a selected set of changes, a revert *removes* a selected set of changes.

A version created by the *Revert* command has two ancestry lines:

- A **dashed blue line** indicates the version from which *Revert* removed some changes.

- A solid black line indicates the *basis version* of the removed patch or change package entry.



See [Reverse Patch: Removing Content Changes](#) on page 246 for more information on how these ancestry lines are created.

## Closest Common Ancestor

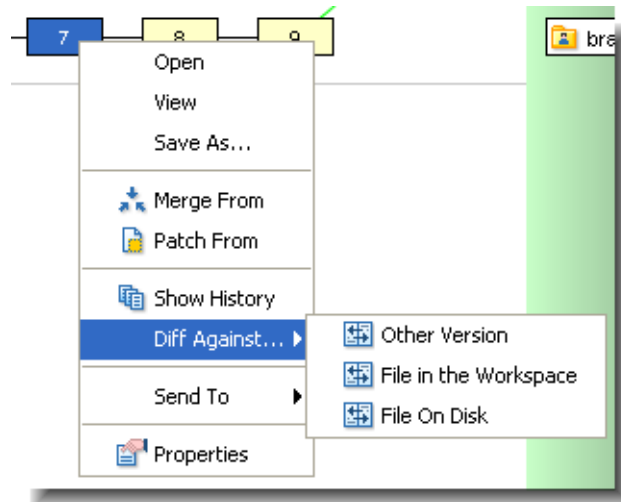
It's instructive to follow all the black and solid-red lines in an element's Version Browser display. This traces the entire ancestry of real versions of an element. In particular, you can use the real-version ancestry to determine the **closest common ancestor** of any two versions. This is the most recent version upon which the two versions are both based, by some combination of direct ancestor and merge connections. (When considering a virtual version in a closest-common-ancestor analysis, first follow the green line back to the corresponding real version.)

You can also use the CLI command `accurev anc -c` to find the closest common ancestor of two versions.

In the *Merge* command, AccuRev determines the closest common ancestor of the two versions to be merged, and uses this version to perform a *3-way merge*. See [The Merge, Patch, and Reverse Patch Algorithms](#) on page 242 for more information on the merge algorithm.

# Operations on Versions

You can perform several operations on a selected version, using its context menu or the Version Browser toolbar.



## Open (equivalent to double-click)

**Windows:** Run the appropriate command on the file, according to its file type. (The Windows file-typing system — "file associations" — does not provide for assigning a file type if the filename has no suffix.)

**UNIX:** Open a text editor on the file.

## View

Open a text editor on a temporary copy of the currently selected file (text files only).

## Save As

Copy the currently selected file to another filename.

## Merge From

Merge the selected version into the version in the workspace from which the Version Browser was invoked.

## Patch From

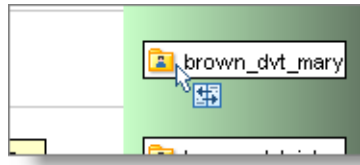
Patch the selected version into the version in the workspace from which the Version Browser was invoked.

## Show History

Open a History Browser tab, showing the transaction that created the selected version.

## Diff Against Other Version

Compare the selected version with another version of the element.



AccuRev changes the mouse pointer, to prompt you to select the other version. You can click on any version in the Version Browser display. You can also click on a stream or workspace label to indicate the version currently in that stream or workspace.

#### **Diff Against File in the Workspace**

Compare the selected version with the file in the workspace from which the Version Browser was invoked. This workspace is listed at the bottom on the AccuRev GUI window. This enables you to invoke a comparison with a file that you have changed in your workspace, but have not yet preserved in the repository with a *Keep* command.

#### **Diff Against File On Disk**

Compare the selected version with the an arbitrary file on your machine. A *File Chooser* dialog appears, in which you specify the file.

#### **Send to Workspace**

Activate the selected element in the workspace from which the Version Browser was invoked. The element must currently have **(backed)** status in the workspace -- that is, it must *not* be active.

#### **Send to Issue**

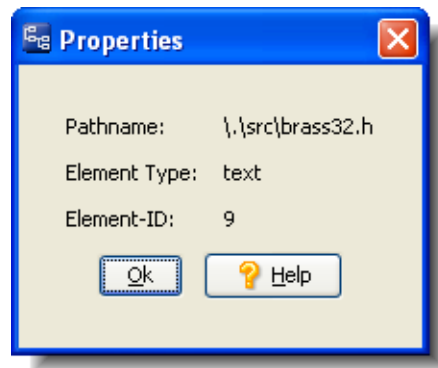
Record the selected version(s) in the change package section (Changes tab) of one or more issue records. The default query is executed, and you are prompted to choose one or more of the records selected by the query. You can also create a new issue record, to which the selected version(s) will be sent. Note: If you have AccuWorkflow enabled, and use a New Issue button from within a dialog in the Java client, you must set any AccuWorkflow fields in those issues manually.

#### **Send to Issue (specifying basis)**

Record the selected version in the change package section (Changes tab) of one or more issue records. You are prompted select a version in the Version Browser display; this version will become the basis version in the change package entry. Then, the default query is executed, and you are prompted to choose one or more of the records selected by the query. You can also create a new issue record, to which the selected versions will be sent. Note: If you have AccuWorkflow enabled, and use a New Issue button from within a dialog in the Java client, you must set any AccuWorkflow fields in those issues manually.

Note: The basis version you specify must be an ancestor of the head version. If it isn't, an "Invalid Change Package" error occurs.

## Properties

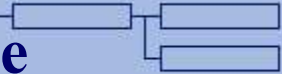


Displays information about the selected element in a pop-up window. The data items displayed vary with the type of element.

## Version Browser Preferences

The *Tools > Preferences* command opens a dialog that includes a *Version Browser tab*, where you can control settings that affect the Version Browser. (See [“AccuRev Preferences \(Tools > Preferences Command\)”](#) on page 42.)

## 6. Diff, Merge, and Change Palette



### The Diff Tool

The AccuRev GUI provides many ways to invoke a "Diff" command to compare two versions of a text-file element. By default, the comparison is performed by AccuRev's own **Diff tool**, which displays the two versions next to each other, using color coding to show the differences and provided convenient navigation controls. You can also configure the GUI to use another file-comparison tool (see [AccuRev Preferences \(Tools > Preferences Command\)](#) on page 42).


### Invoking the Diff Tool (or Another File-Comparison Tool)

No matter which tool you've configured to perform file comparisons, you invoke the tool in the same way, as described below. AccuRev's Diff tool opens in a separate tab within the AccuRev GUI window; any other tool opens in its own window.

There are multiple ways to specify the two versions of a file element to be compared:

In a File Browser, each file's context menu includes a *Diff Against...* item. There are several choices, each of which compares your version (i.e. the file in your workspace) with a version stored in the depot:



- **Most Recent Version:** Compares your file with the version currently in your workspace stream. Use this choice if you've modified the file since the last time you performed a Keep on it (or if you've never performed a Keep on it since the last update).
- **Backed Version:** Compares your file with the version currently in the backing stream. For example, you might use this choice to see all the changes you've made to this file since you updated your workspace and starting working on the file. (And assuming no one else has promoted a new version to the backing stream in the meantime.) This might include the changes stored in several intermediate versions that you've created with keep.
- **Basis Version:** Compares your file with the version that you started working with, before making your "recent" changes. For a discussion of the meaning of "recent", see [Patches and Change Packages](#) on page 172.
- **File on Disk:** Compares your file with an arbitrary file on your machine. A standard operating system "open file" dialog box open, in which you specify the file.

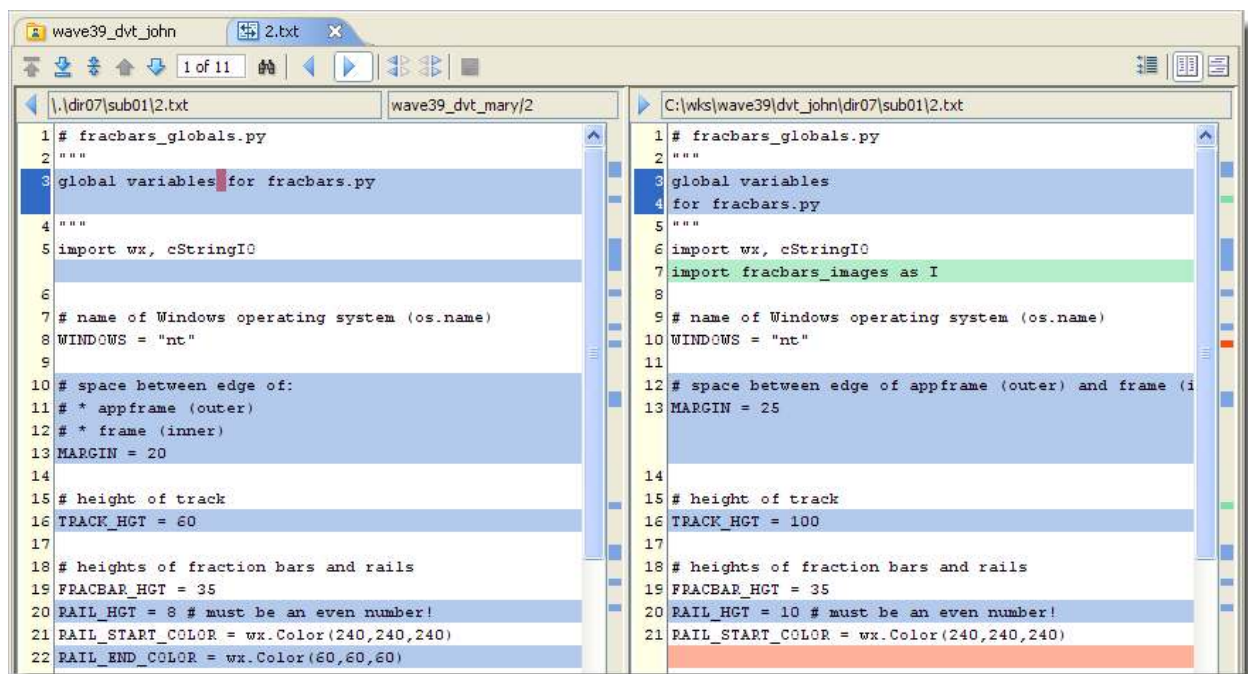
In a Stream Browser display, view the elements in a stream's *default group* by clicking the  control below the stream. An element's context menu includes the *Diff Against...* items described above.

In a Version Browser or Stream Version Browser display for a text-file element, you can compare any two versions: right-click any version, select *Diff Against Other Version* from the context menu, then left-click any other version.

In a History Browser display for a text-file element, you can compare the versions created in any two transactions. Right-click a transaction in the Summary (upper) pane, select *Diff Against Other Version* from the context menu, then left-click another transaction in the Summary pane. Similarly, the *Diff Against File in the Workspace* command enables you to compare the version created in a particular transaction with the file currently in your workspace. (That file might contain changes that you haven't yet preserved with *Keep*.)

## Diff Tab Layout

The Diff tool displays the two versions side-by-side in separate panes, so that corresponding text lines in the versions line up visually. (You can use buttons  *Horizontal Layout* and  *Vertical Layout* at the right side of the toolbar to arrange the panes above-and-below.) We'll call them the "before version" (displayed on the left, or above) and the "after version". The element pathname and the version-IDs are displayed above the versions' contents.



Initially, the two panes are the same width, but you can drag the vertical separator to change the relative widths. To make both panes wider, just increase the size of the overall AccuRev GUI window.

### Notes:

- Namespace differences

The two versions being compared can have different pathnames: the element may have been renamed or moved to another directory within the depot. (This may have occurred in

one of the versions, or in both of them.) The Diff tool does not highlight or announce such *namespace differences*.



- Left vs. right

Depending on how you launch the Diff tool, an older version might be displayed on the right, and a newer version on the left. We recommend keeping the older version on the left, so that "before" and "after" in the descriptions below correspond to reality.

- Comparing binary files

If the versions to be compared are in a binary image format that AccuRev can render, the Diff tool simply displays the versions, so that you can determine their differences by inspection. AccuRev can render the following image formats:

- JPEG (.jpg or .JPG filename suffix)
- PNG (.png or .PNG)
- GIF (.gif or .GIF)

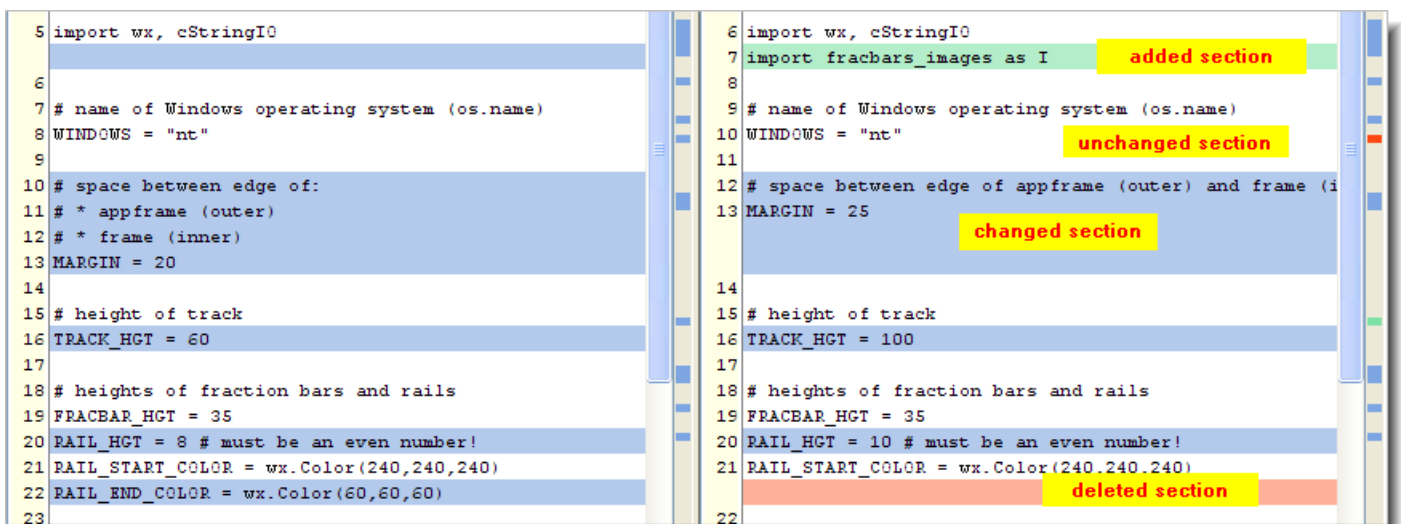
- Applying changes from the "before" version to your workspace's version

If you're comparing the file in your workspace with another version, you can use the Diff tool to edit your file as you view the differences.

See [Editing a File Using the Diff Tool](#) on page 212.

## Difference Section Color-Coding

Through color-coding, the Diff tool partitions the text into the following kinds of "difference sections":



- *Unchanged Section* (white background): Text section in which the versions are identical.
- *Added Section* (green background): Text section that occurs only in the "after" version. Empty space with a blue background appears at that point in the "before" version.
- *Deleted Section* (red background): Text section that occurs only in the "before" version. The deleted section appears with a blue background in the "before" version.
- *Changed Section* (blue background): Sometimes, the Diff tool decides that a text section in the "before" version has been revised, producing the corresponding section in the "after" version. The before and after sections are not necessarily the same length, in which case some empty space is displayed in one of the versions. Both the before and after sections appear with a blue background. Both sections contain red character highlights, indicating the first location in a line where the two versions differ:

```

1 def get_topdir():
2     if 'ACCUREV_BIN' in os.environ:
3         if = os.popen(os.environ['ACCUREV_BIN'])
4     else:
5         f = os.system('accurev info')

```

```

1 def get_topdir():
2     if 'AC_BINDIR' in os.environ:
3         if = os.popen(os.environ['AC_BINDIR'])
4     elif:
5         f = os.system('accurev info')

```

If a text block has been moved from one location in the file to another, the Diff tool indicates this as two separate changes: a deleted section at the original location, and an added section at the new location.

Your user preferences include controls for the Diff tool's handling of whitespace-only differences between the versions. (See [AccuRev Preferences \(Tools > Preferences Command\)](#) on page 42.)

## Navigating in a Diff Tab

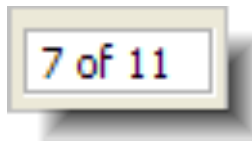
Most text files are too long to fit on the display screen, and some individual text lines are too wide. Accordingly, the panes in which the versions appear have both vertical and horizontal scroll bars. Scrolling of the panes is synchronized automatically. Line numbers appear at the left edge of each pane.

For a text file that contains hundreds or thousands of lines, there may be only a few difference sections (added, deleted, or revised), separated by large unchanged sections. The Diff tool provides both navigation buttons on its toolbar and a difference map for direct access to a particular difference section.

### Diff Toolbar Navigation Buttons



Whenever you use one of these buttons to jump to a particular difference section, the Diff tool remembers it as the *current difference* and highlights the lines numbers in both panes.



A "difference N of M" window in the toolbar indicates which difference section is current.

#### **First Diff**

(Keyboard shortcut: **Ctrl-F**) Go to the first difference section.

#### **Last Diff**

(Keyboard shortcut: **Ctrl-L**) Go to the last difference section.

#### **Center Diff**

Using the scroll bars or the difference map, you can make the *current difference* scroll offscreen. Click this button to bring it back onscreen.

#### **Previous Diff**

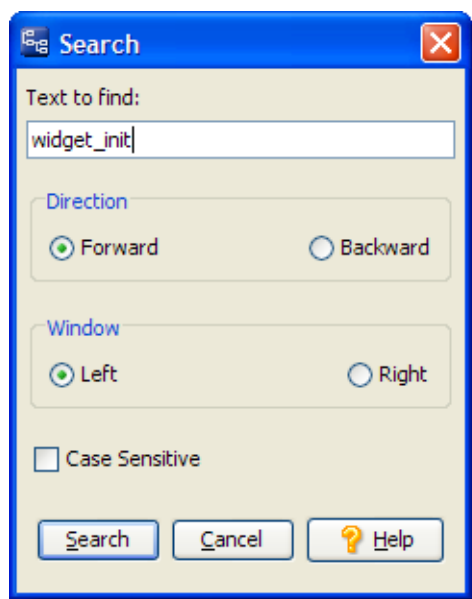
(Keyboard shortcut: **Ctrl-P**) Go to the previous difference section.

#### **Next Diff**

(Keyboard shortcut: **Ctrl-N**) Go to the next difference section.

#### **Search**

(Keyboard shortcut: **Ctrl-S**) Search for a text string, in either of the versions being compared. A dialog appears, in which you define the search.

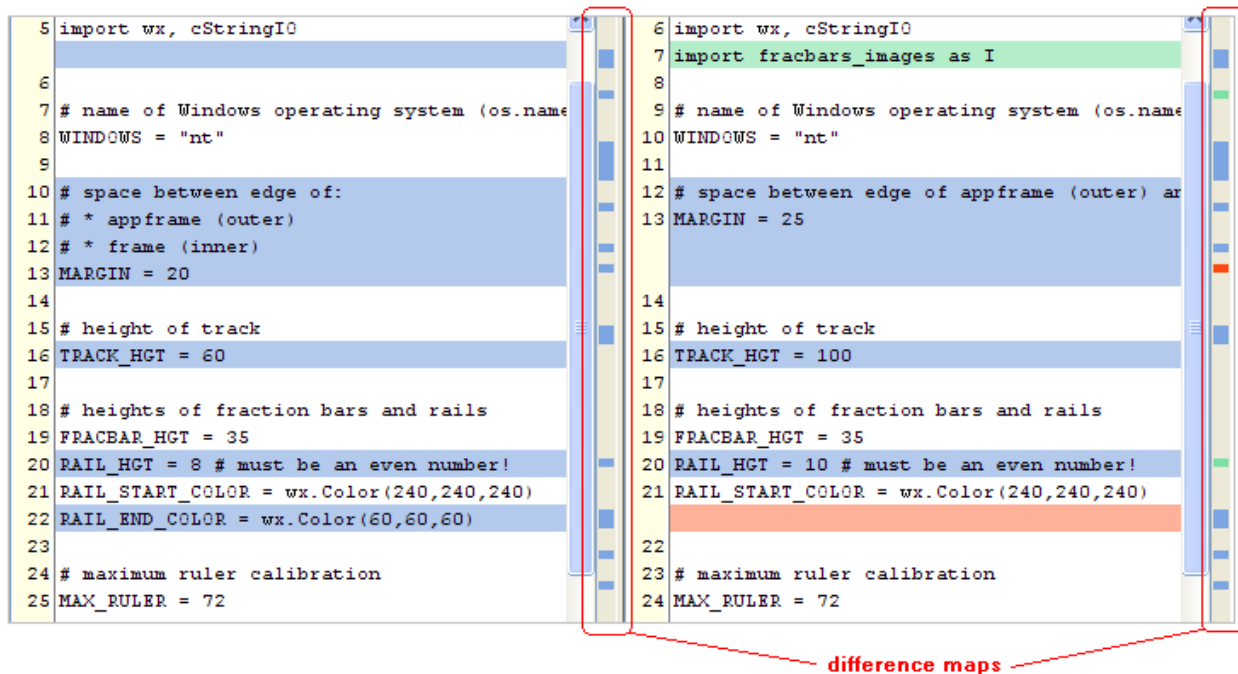


This command is also available on the GUI window's main menu: *File > Search*.


After you've defined a search, you can perform it again, using the command *File > Search Again* on the GUI window's main menu, or its keyboard shortcut: function key **F3**.

## Difference Maps

To the right of each pane, there's a difference map that shows the relative locations and sizes of all the difference sections. The maps use the same color-coding (see [Color-Coding of the Contributor Versions](#) on page 232) as the difference sections themselves. Click on any colored area within a map to scroll both panes directly to the corresponding difference section. (Actually, you can click anywhere in the difference map; the panes will scroll to that location, even if the files are identical there.)



Using a difference map merely scrolls the panes; it does not affect which difference is the current difference. After you jump to a particular location, click the difference section in either pane to make it the current difference.


If you use a difference map, but then want to scroll back to where you were before, click the  *Center Diff* toolbar button.

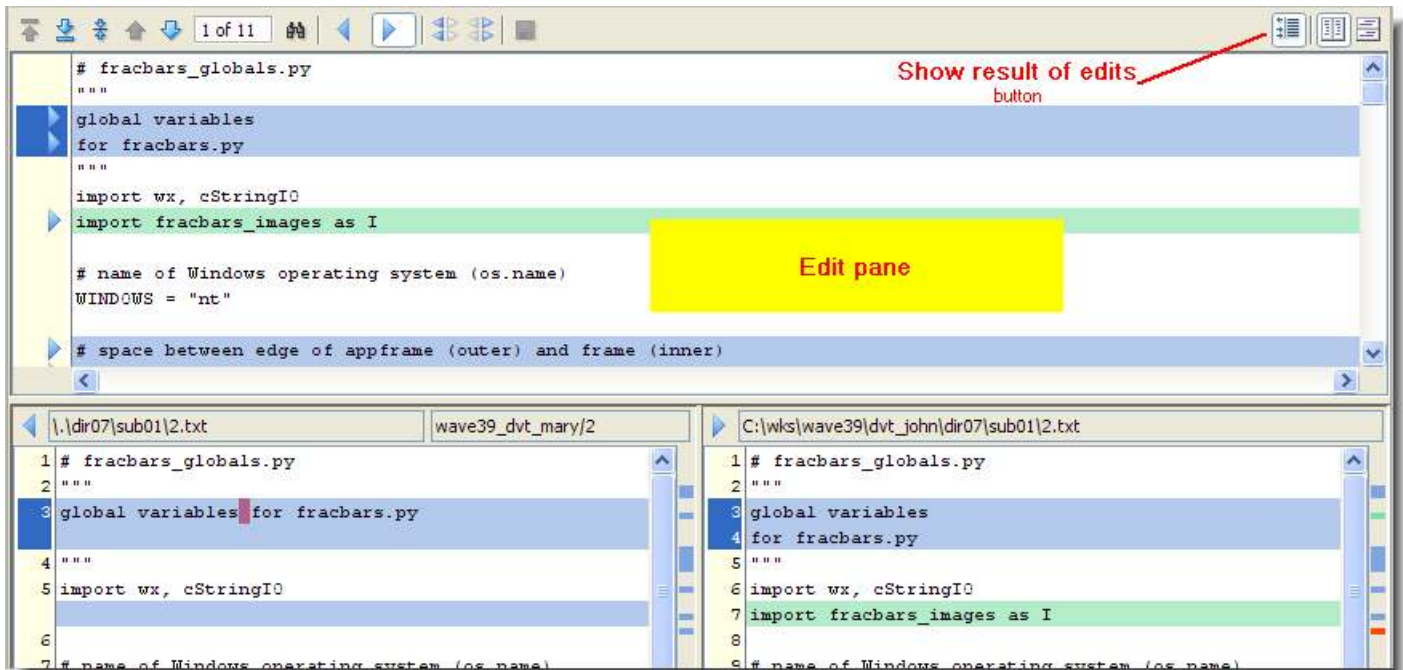
## Editing a File Using the Diff Tool




In addition to comparing two versions of a file, the Diff tool can help you to edit the contents of the version in your workspace. This "edit-by-diff" capability is available only when you're comparing your workspace version with another version; it's not available when you invoke the Diff tool from the Version Browser or History Browser.

Here's a procedure for using this capability (see the toolbar button summary below):

1. Invoke the Diff tool in a File Browser, to compare your workspace version with some other ("before") version of the same file.

- Click the  Show result of edits button in the Diff toolbar. This opens a third pane (the Edit pane), which initially contains the same text as your workspace version.







- As you browse through the difference sections, you can:
  - Edit text manually in the Edit pane.
  - Click the  Revert my change toolbar button to swap in the "before" version's text at that point (a change, addition, or deletion). The button becomes highlighted to indicate your selection.
  - Include both the "before" and "after" versions' text at that point: click the  Revert My Change toolbar button, then hold down the Ctrl key and click the  Restore my change toolbar button.




- Alternatively, click the two buttons in the opposite order. Both buttons become highlighted, and number annotations indicate the order you selected:

You can then modify the combined text manually.

- Change your mind about a difference section where you've used  and/or , by selecting that section and clicking one of those buttons. Any manual edits you've made in that difference section are lost.
- Click the  Revert all of my changes button to make the Edit pane the same as the "before" version.
- Click the  Restore all of my changes button to make the Edit pane the same as the "after" version.

- Finish your work in either of these ways:

- Click the  Save edits and close toolbar button. This simply replaces the file in your workspace. (It doesn't perform an AccuRev Keep command.)
- Close the Diff tab to cancel the edit-by-diff session. No changes are saved to the file in your workspace.

*Note: Edit-by-Diff vs. Merge*



Using the edit-by-diff capability is similar to using the *Merge Tool* (see [The Merge Tool](#) on page 229). But there's an important difference:

- An edit-by-diff operation involves just two versions.
- A merge operation takes into account a third version: the closest common ancestor of the two versions being merged.



## Diff Toolbar Edit-by-Diff Buttons



### Revert my change

Use the "before" version's section at this point. If the  button is currently highlighted, you can hold down the **Ctrl** key and press  to append the "before" section to the workspace version's section.

### Restore my change

Use the workspace version's section at this point. If the  button is currently highlighted, you can hold down the **Ctrl** key and press  to append the workspace version's section to the "before" section.

### Revert all my changes

Make the Edit pane the same as the "before" version.

### Restore all my changes

Make the Edit pane the same as the "after" version.

### Save edits and close

Store the current contents of the Edit pane in the workspace's file, without performing a *Keep* operation.

## Main Menu Commands Available in a Diff Tab

Several of the commands in the *Edit* submenu of GUI window's main menu are enabled when you're working in a *Diff* tab:

### Cut

(Keyboard shortcut: **Ctrl-X**) Cut the selected text to the window system's clipboard.



### Copy

(Keyboard shortcut: **Ctrl-C**) Copy the selected text to the window system's clipboard.



### Paste

(Keyboard shortcut: **Ctrl-V**) Paste the contents of the window system's clipboard.

### Select All

(Keyboard shortcut: **Ctrl-A**) Select the entire contents of the current pane.



### Search

(Keyboard shortcut: **Ctrl-S**) See the *Search* description above.



### Search Again

(Keyboard shortcut: **F3**) See the *Search* description above.

## The Change Palette Tool

The Change Palette is a tool for propagating changes 'outside the lines' of a depot's *stream hierarchy*. You use it to *promote* one or more versions from a *dynamic stream* to a stream other than its *parent stream*. In some cases, a *merge* operation is required before you can propagate changes between streams; the Change Palette helps you to manage the *Merge* and subsequent *Promote* commands. You can use *Patch* instead of *Merge*, in order to send just some -- not all -- of the changes that a version contains.

You can also use the Change Palette to send changes from a dynamic stream to a workspace. This can be accomplished with a simple *checkout* operation, or with a more complex merge or *patch* operation.

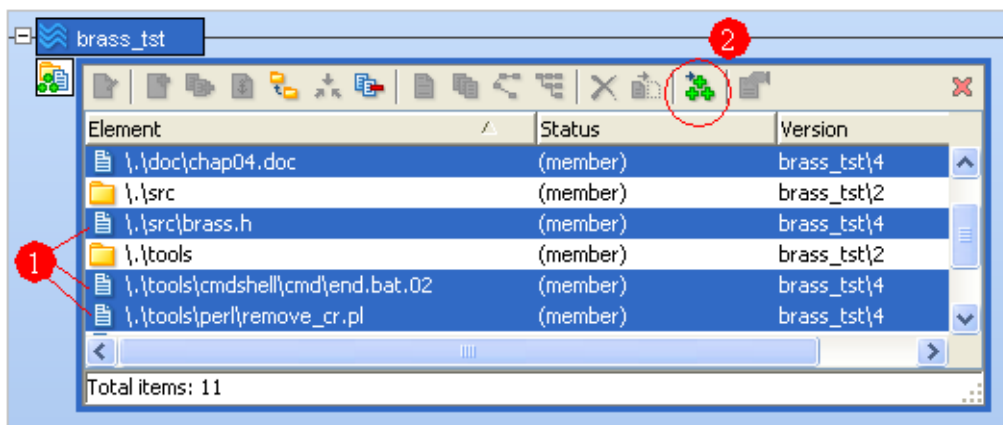
You cannot use the Change Palette to promote versions from your workspace to an arbitrary stream. The only destination for versions in a workspace is the workspace's backing stream, via the *Promote* command. (But you can *reparent* a workspace to provide a new destination for promoted versions.)


## Opening a Change Palette Tab

Before using the Change Palette itself, you must specify the version(s) that are to be promoted 'outside the lines' of the depot's stream hierarchy. You can do this in the variety of contexts listed below.

### ***From a StreamBrowser: Selected items from a current development activity subwindow***

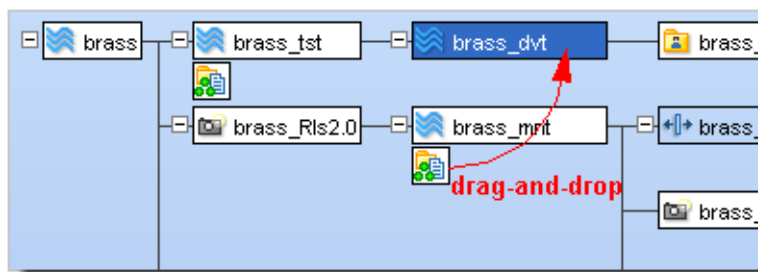
The current development activity subwindow shows the *development activity* currently taking place in a stream or workspace -- *by element, by transaction, or by issue record*. In this subwindow, you can:



1. Select any number of items (elements, transactions, or issue records).
2. Invoke the Send to Change Palette command, using the  toolbar button or the selection's context menu.

### ***From a StreamBrowser: All the versions with changes that have not been propagated to the destination***

Instead of selecting individual versions, you can let AccuRev determine the complete set of versions in a source stream that have not yet been propagated to a destination stream or workspace.

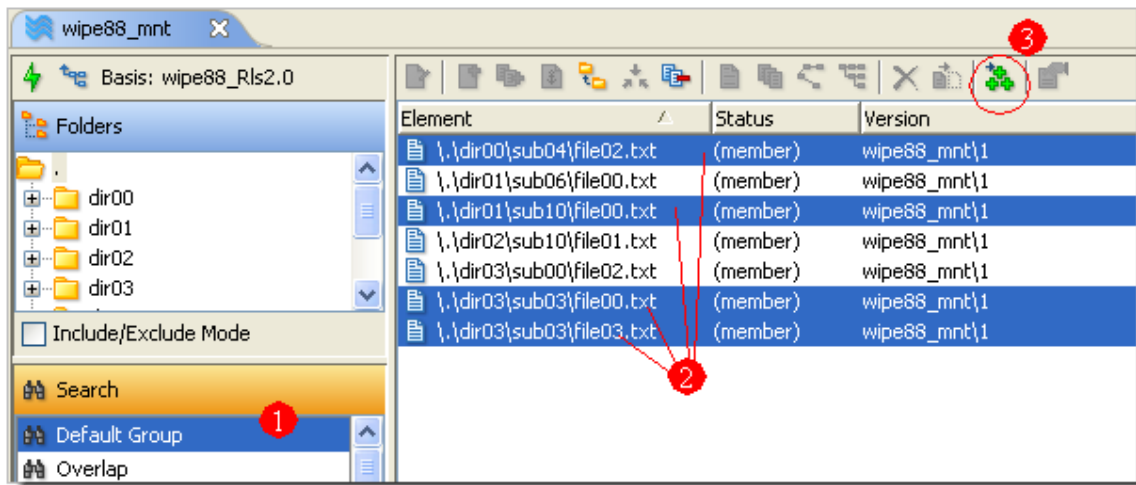



This can include versions that are not currently in the source stream's default group at all, but are inherited from a higher-level stream.

Click the control below the source stream, and drag it to the destination stream or workspace.

***From the Details pane of a File Browser, opened on a stream: Selected versions from the default group***

This is essentially the same as working in a Default Group subwindow in the StreamBrowser:

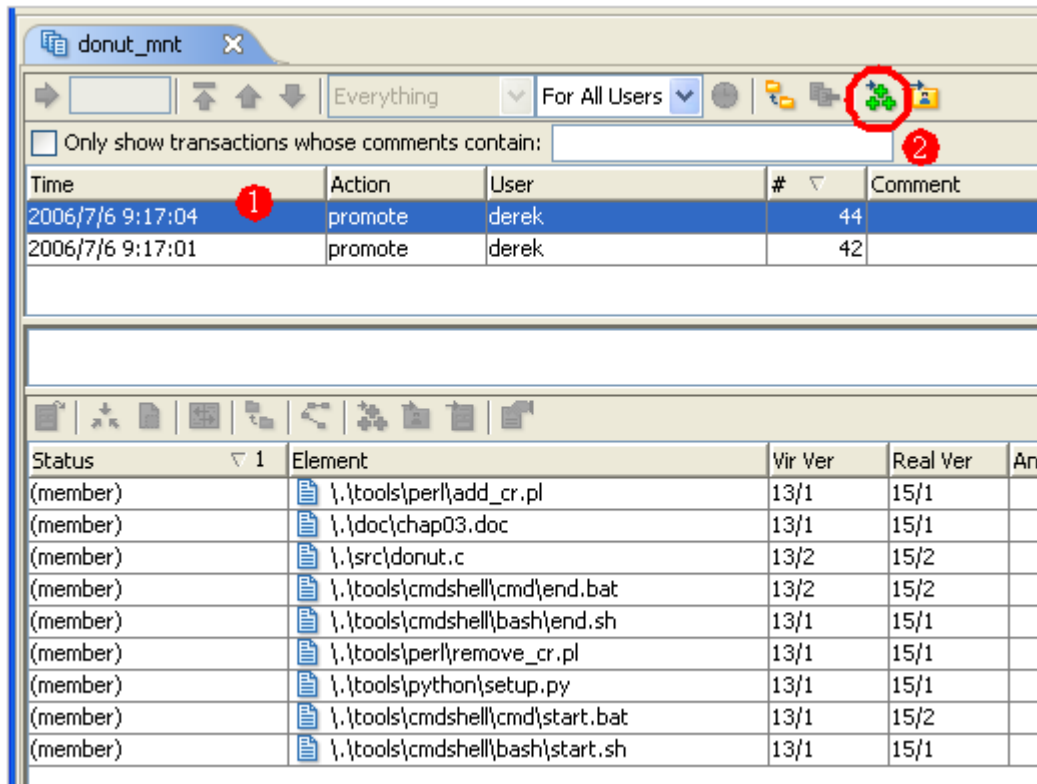



1. Select the Default Group search.
2. Select any number of the stream's active versions.
3. Invoke the *Send to Change Palette* command, using the  toolbar button or the selection's context menu.

***From a History Browser or Active Transactions tab, opened on a stream: All the versions in specified transaction(s)***

A History Browser tab shows all the transactions that created (via *Promote*) versions in a stream. An *Active Transactions* tab shows the subset of these transactions with at least one

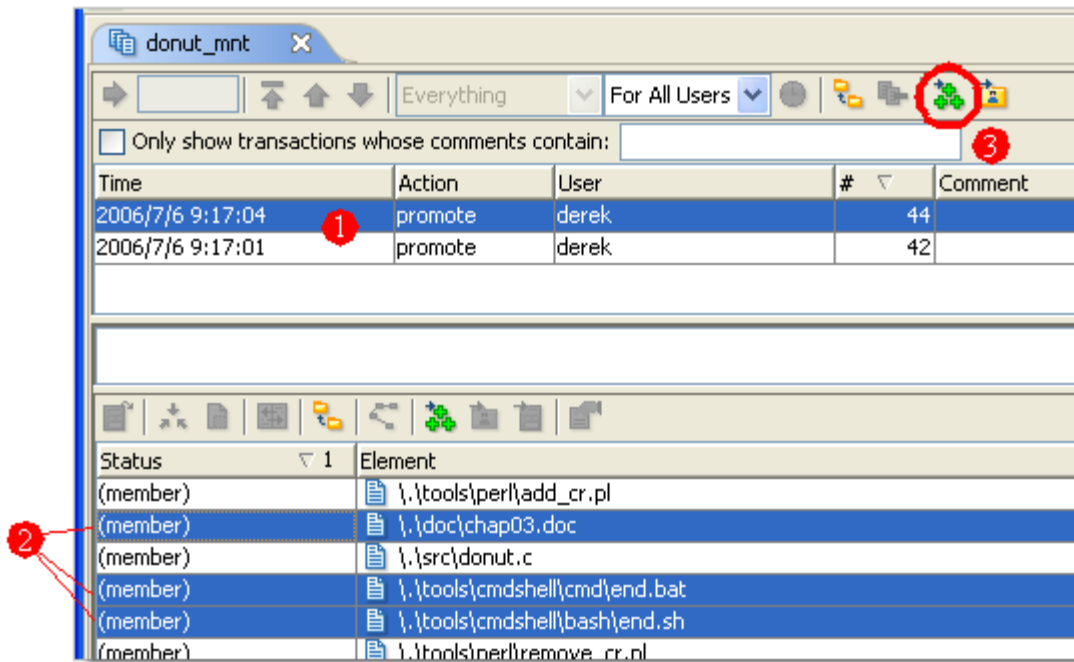
version still active in the stream. In either one, you can load a Change Palette with the versions created one or more selected transactions.




1. Select one or more transactions in the Summary (upper) pane.
2. Invoke the Send to Change Palette command, using the  toolbar button or the selection's context menu.

***From a History Browser or Active Transactions tab, opened on a stream: Specified version(s) from one transaction***

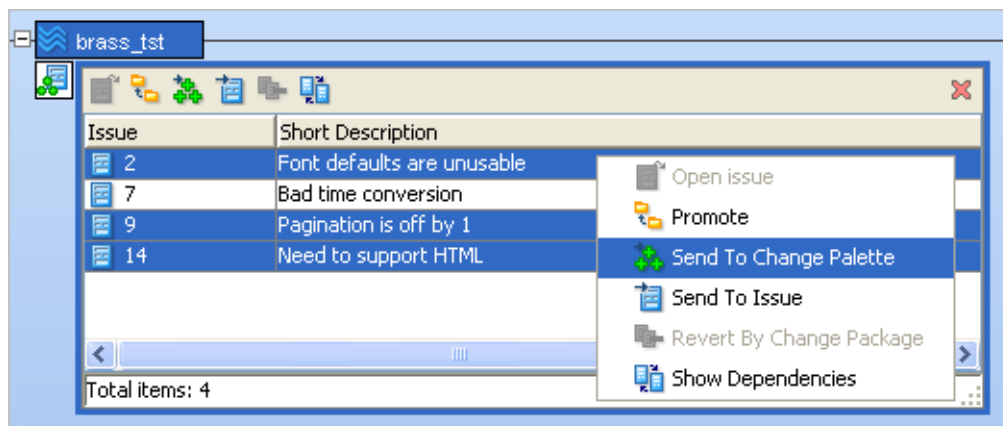
This is similar to the previous case: instead of loading a Change Palette with *all* the versions created in a transaction, you load selected versions from a transaction.




1. Select one transaction in the Summary (upper) pane.
2. Select any number of that transaction's versions in the Versions (lower) pane.
3. Invoke the Send to Change Palette command, using the  toolbar button or the selection's context menu.

***From a Stream Issues tab, opened on a stream: All the versions in the change package(s) of the specified issue record(s)***

A Stream Issues tab shows the issue records whose change packages are "in" a particular stream.



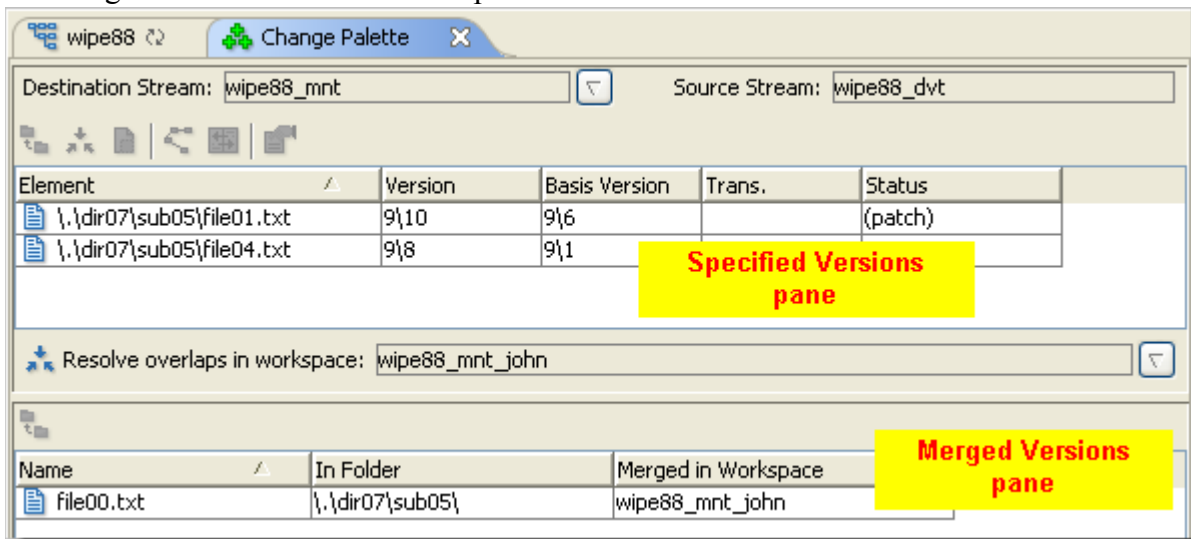
Each change package contains a "patch" of data from one or more text-file elements. You can load a Change Palette tab with the versions in one or more issue records' change packages:

1. Select one or more issue records in the Issues (upper) pane.
2. Invoke the Send to Change Palette command, using the  toolbar button or the selection's context menu.

In all cases, a new Change Palette tab appears, loaded with the versions you've specified.

## Change Palette Tab Layout

The Change Palette tab includes these panes:







- The *Specified Versions* pane initially lists all the versions you specified with the *Send to Change Palette* command (or the drag-and-drop operation). The versions' entries are cleared from this pane as you work with them using *Promote*, *Merge*, *Patch*, etc.
- The *Merged Versions* pane appears only after you've merged or patched versions in the Specified Versions pane. It lists the versions created by the *Merge* or *Patch* command, and provides a convenient way to *Promote* these versions to the destination stream.

### Layout of the Specified Versions Pane

This pane contains a table that initially lists all the versions loaded into the Change Palette.

Destination Stream:  Source Stream: wipe88\_dvt



Element	Version	Basis Version	Trans.	Status
 \dir07\sub05\file00.txt	9/7	9/1		
 \dir07\sub05\file01.txt	9/10	9/6		
 \dir07\sub05\file04.txt	9/8	9/1		

As you work with these versions, they "migrate" out of this pane:

- Entries for versions that you promote to a destination stream -- or send to a destination workspace -- are automatically cleared from the Change Palette.
- Entries for versions that you *Patch* into a workspace (based on the source or destination stream) are also cleared automatically.
- Entries for versions that you *Merge* into a workspace (based on the source or destination stream) are automatically cleared from the Specified Versions pane. Corresponding entries, representing the versions created by the *Merge*, are added to the Merged Versions pane.

## Columns in the Specified Versions Table

The Specified Versions table includes these columns:

### Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

### Version

The version of the element that was sent to the Change Palette.

### Basis Version

The version/basis-version pair defines the "patch data" that will be incorporated into the destination version if you invoke the *Patch* command. (See [Patches and Change Packages](#) on page 172.)

### Transaction

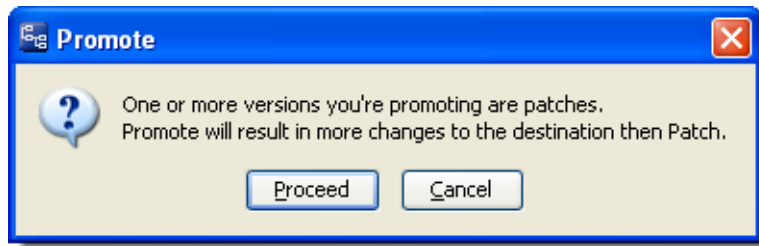
(if you invoked *Send to Change Palette* from a History Browser or *Active Transactions* tab) The number of the transaction from which you sent this version.

### Status

The relationship between the version in the source stream and the version in the destination stream:

- **(overlap)** indicates that you must perform a *Merge* command prior to *Promote*'ing this element to the destination stream. You can also perform a *Patch* command to propagate *some* of this version's changes to the destination stream, but this does not resolve the *overlap* status (and so does not enable the *Promote* command).
- **(patch)** indicates that the change that you sent to the Change Palette for this element (as specified by its entries in the Version and Basis Version columns) is a patch. That is, the change does *not* include the entire set of changes between the source-stream version and the closest common ancestor of the source-stream and destination-stream versions. You can perform a *Patch* command to propagate the changes in the patch to the destination stream.

If the element does not also have (**overlap**) status, the *Promote* command is enabled for this version, too.



But promoting would propagate more changes to the destination stream than were loaded into the Change Palette. AccuRev issues a warning if you invoke *Promote* on this version.

- A blank indicates that there is no overlap and that the change loaded into the Change Palette is *not* a patch. This means you can safely *Promote* the version to the destination stream.

### Other Indicators and Controls

The Specified Versions pane also includes:

#### Source Stream

The name of the stream from which you sent the versions, using *Send to Change Palette* or the drag-and-drop operation.

#### Destination Stream

Initially blank (unless you used the drag-and-drop operation to load the Change Palette). Subsequently, it displays the name of the stream or workspace you've selected as the versions' destination.

#### Resolve overlaps in workspace

This field does not appear until the first time you invoke the *Merge* command. At that point, AccuRev prompts you to specify a workspace for the merge operation to take place in. It then displays the workspace's name in this field (which is also a listbox control for changing the name).

### Layout of the Merged Versions Pane

This pane contains a table that lists the versions you've created by invoking the *Merge* command on versions in the Specified Versions pane. As you proceed to *Promote* these newly created versions, their entries are cleared from the Change Palette.

A screenshot of a table within a software interface. The table has three columns: "Name", "In Folder", and "Merged in Workspace". There are two rows of data. The first row shows "bagel.h" in the Name column, ".\src\" in the In Folder column, and "bagel\_dvt\_john" in the Merged in Workspace column. The second row shows "commands.c" in the Name column, ".\src\" in the In Folder column, and "bagel\_dvt\_john" in the Merged in Workspace column. The table is set against a light beige background with a thin border.

Name	In Folder	Merged in Workspace
bagel.h	.\src\	bagel_dvt_john
commands.c	.\src\	bagel_dvt_john

The Merged Versions table includes these columns:

### Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

### Merged in Workspace

The name of the workspace in which the merged version was created. This is not necessarily the same for all the listed elements, since you can switch workspaces using the *Resolve overlaps in workspace* control.

## Working in a Change Palette Tab

The following sections describe the tasks you perform in a Change Palette tab. The order of the sections follows the typical order of tasks. But the Change Palette is a long-lived "work environment", not a simple dialog box or sequential wizard. So you might perform just some of the tasks and/or perform them in a different order.

### Specifying the Destination Stream

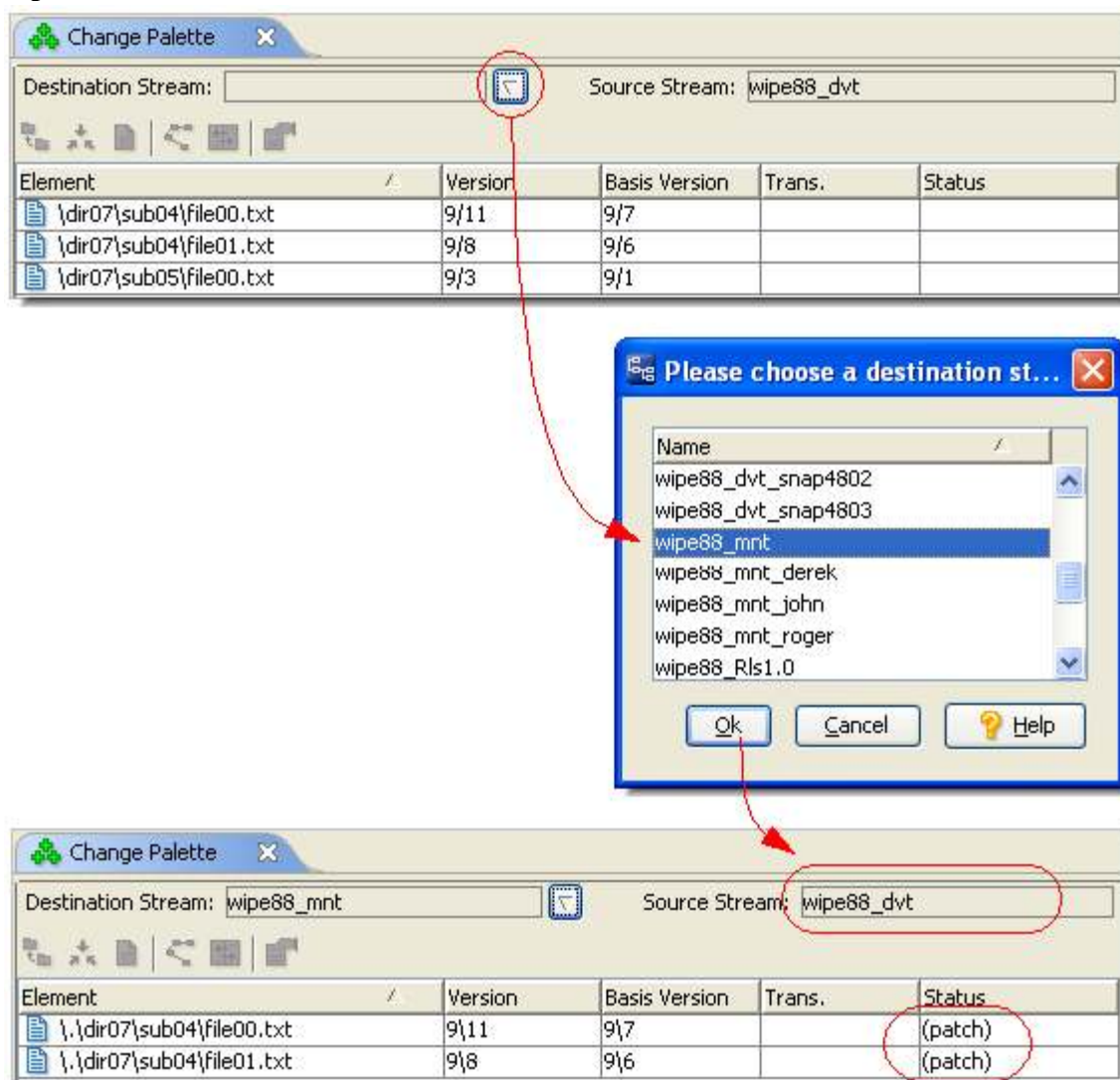
If you've loaded versions into the Change Palette using the *Send to Change Palette* command, you've already indicated the source stream, but need to specify the destination stream or workspace. (The drag-and-drop operation indicates both the source and destination.)

Click the arrow control next to the Destination Stream field.



A dialog appears, listing all of the depot's streams and workspaces. Select one to be the destination for subsequent commands (**Send to Workspace**, **Promote**, **Merge**, and/or **Patch**) on the versions in the Specified Versions pane.

## Example



As the example shows, AccuRev determines a *status* for each element, describing the relationship between the source-stream version and the destination-stream version. Entries with **(overlap)** status are highlighted in yellow.


Note: If the destination is a workspace, AccuRev evaluates the version currently in the workspace stream, which is not necessarily the same as the file currently in the workspace tree.

You can change the destination stream at any time, using the arrow control next to the Destination Stream field.

## Sending Versions to the Destination Stream

In the Specified Versions pane:

- If the destination is a stream, you can immediately *Promote* the versions that do not have **(overlap)** status.
- If the destination is a workspace, you can immediately *Send to Workspace* any or all versions, whether or not they have **(overlap)** status.

Select one or more eligible versions, then click the  button in the pane's toolbar or choose *Promote* (or *Send to Workspace*) from the selection's context menu. When the command completes, the entry(s) for the version(s) are cleared from the Change Palette.

In the Merged Versions pane, you can *Promote* versions at any time. The versions are active in the workspace listed in the Merged in Workspace column; *Promote* sends them to this workspace's backing stream.


*Notes:*

- If you've used multiple workspaces for merging ...  
In each *Promote* invocation, the set of versions must all be in the same workspace.
- If you've merged using a workspace based on the source stream ...  
Promoting sends the merged version to the workspace's backing stream, which is not the destination stream you originally specified. You can now promote this version from the workspace's backing stream to the original destination stream (no additional merge is required). But you'll need to use another instance of the Change Palette to invoke the *Promote* command.

## Merging the Source and Destination Versions

If an entry in the Specified Versions pane has **(overlap)** status, you cannot immediately promote the version to the destination stream. First, you must create a new version by merging the source and destination versions.

The merge operation must take place in a workspace that belongs to you and is based on either the source or destination stream. If you have not already selected a merge workspace, you must do so before performing any merges. (See [Merging the Source and Destination Versions](#) on page 225.)

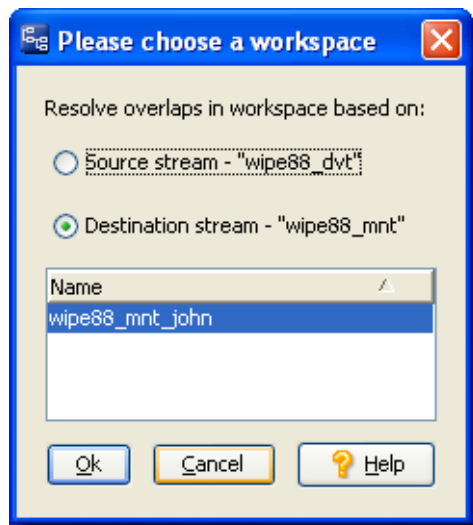
Select one or more versions with **(overlap)** status, and click the  button in the pane's toolbar or choose *Merge* from the selection's context menu.

The merge operation(s) proceed for the selected element(s), as described in [The Merge Command](#) on page 229..

Note: If the merge workspace was not up-to-date, it's possible that the element will continue to have (overlap) status after the merge. In this case, perform a standard **Merge** in the workspace before promoting the element.

## Selecting a Workspace for Performing Merges

A merged version cannot be created directly in either the source or destination stream, because these are *dynamic streams* — all new versions of AccuRev elements must be originally created in workspaces.

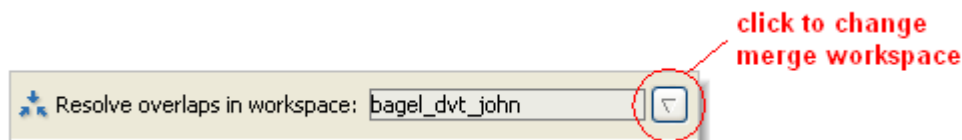


The first time you invoke *Merge* in the Specified Versions pane, AccuRev prompts you to establish a merge workspace, suggesting one that is based on the destination stream. You can also choose a workspace based on the source stream.


### Notes:

- Which is preferable -- source or destination?  
A workspace based on the destination stream is preferable. After you perform the merge, you can then accomplish the original task -- propagating a version's changes to the destination stream -- with a single *Promote*. If you use a workspace based on the source stream, you'll need to *Promote* twice: (1) from the merge workspace to the source stream; (2) from the source stream to the destination stream. The second one requires you to use another instance of the Change Palette.
- What if no candidate workspace exists?  
Click *Cancel* in the choose-workspace dialog, and do either of the following:
  - Create a workspace based on one of the streams.
  - Reparent one of your existing workspaces to the source or destination stream. Be sure to Update the reparented workspace before using it to perform merges in the Change Palette.

After you select a merge workspace, its name is displayed at the bottom of the Specified Versions pane. You can change this setting at any time; this enables you to use different workspaces to merge different versions.



## Patching Changes from the Source Version into the Destination Version

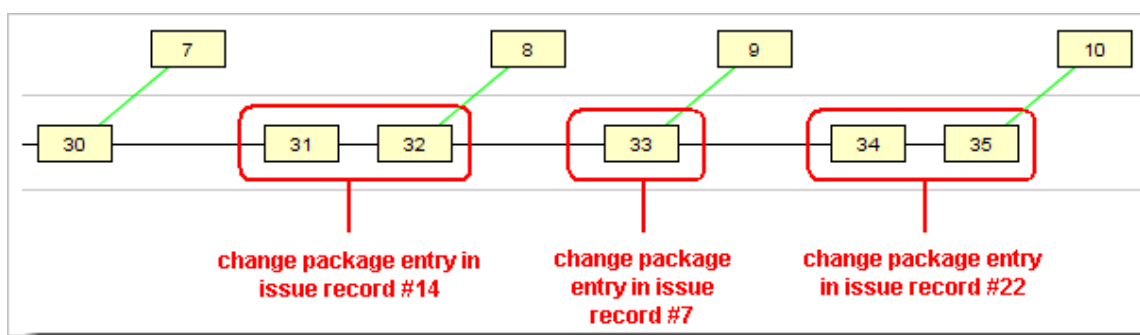
If an entry in the Specified Versions pane has **(patch)** status, you can invoke the *Patch* command to incorporate the patch defined by the Version and Basis Version settings into the destination-stream version. Select one or more of these versions, and click the  button in the pane's toolbar or choose *Patch* from the selection's context menu. AccuRev prompts you to select a workspace that belongs to you and is based on the destination stream. The patch operation is performed in this workspace, as described in [Patches and Change Packages](#) on page 172.

When the patch operation is complete, the entry disappears from the Specified Versions pane.



### Multiple Entries for the Same Element

In some cases, the Specified Versions pane of the Change Palette can get populated with multiple entries for the same element. Here is a simple scenario:

1. Successive changes to an element, *brass.c*, get assigned to different AccuWork issue records. For example, this illustration shows a two-version change assigned to issue #14, then a one-version change assigned to issue #7, then a two-version change assigned to issue #22.



2. At this point, all three issues (#14, #7, and #22) are "in" the stream to which the versions were promoted.
3. Open a Stream Issues window for the stream to which the versions were promoted, then select issues #14 and #22 and invoke the Send to Change Palette command. Since there is a "change package gap", consisting of version 33, between the two change package entries for *brass.c*, AccuRev cannot combine them into a single patch entry. Accordingly, the two entries are entered into the Change Palette separately:

Element	Version	Basis Version	Trans.	Status	Count
 \\src\brass.c	9\35	9\33		(patch)	2
 \\src\brass.c	9\32	9\30		(patch)	2

The total number of entries for a given element is displayed in the Count column. The (patch) indicators in the Status column indicate that Patch is the appropriate command to invoke on each entry, in order to propagate the changes to *brass.c* in issue records #14 and #22 (and no other changes) to the destination stream.

## Commands Available in a Change Palette Tab

### Clear

Remove the selected entry(s). You can clear entries from either pane. This capability helps you to keep track of your work (and the work that you've decided not to do).



If you attempt to close the Change Palette tab when either pane is non-empty, AccuRev prompts you for confirmation. **Note:** **Clear** is available from the right mouse button context menu; it is not currently available from the tool bar.

### **Promote**

(available only if the destination is a dynamic stream)

Specified Versions pane: promote the selected version(s) to the destination stream.

Merged Versions pane: promote the selected version(s) from the workspace listed in the Merged in Workspace column to its backing stream.

### **Send to Workspace**

(available only if the destination is a workspace) Performs a *Send to Workspace* command on the selected version(s) in the destination workspace. (see [The Send to Workspace Command](#) on page 132.)

### **Merge**

Merge the selected version(s) with the corresponding version(s) in the destination stream, creating the merged version(s) in the workspace listed in the *Resolve overlaps in workspace* field. When the command finishes, the entry(s) are cleared from the Specified Versions pane, and entry(s) for the newly created version(s) are added to the Merged Versions pane.

### **Patch**

Performs a [Patch](#) command from each selected version to the corresponding version in the destination workspace, or in a workspace that you select below the destination stream. When the command finishes, the entry(s) are cleared from the Specified Versions pane.

### **Browse Versions**

Open a *Version Browser* tab, showing all the versions of the selected element, and their interrelationships (ancestry).

### **Diff**

Compares the selected version with the version in the destination stream or workspace.

Note: When the destination is a workspace, the comparison is performed with the version in the workspace stream, not the file in the workspace tree. These are different if the element has (modified) status in the workspace.

## Properties

Displays information about the selected element. The data items displayed vary with the type of element.

# The Merge Tool

## The Merge Command

The *Merge* command combines two versions of an element, producing a new version. (That is, a successful merge operation always ends with a **Keep**. This is a software configuration management best practice.) AccuRev can process both *content* changes and *namespace* changes in performing a merge. By default, the merging of text-file contents is performed by AccuRev's own Merge tool, which uses a 3-way merge algorithm (see ).

## What about elements that aren't text files?

The Merge command can also handle any other kind of version-controlled element: *binary file*, *directory*, or *link*.

## Configuring the GUI to use another text-file-merge tool

A *user preference* enables you to configure a third-party tool for merging the contents of text files.

## Why Do I Need to Merge?

Perhaps the most common usage pattern in AccuRev is:

- Using *Keep* to create one or more versions of a text file in your workspace.
- Using *Promote* to propagate the most recently kept version to the *backing stream*.

Occasionally, someone else "gets there first". That is, both you and a colleague are working on the same file concurrently, each of you using a copy of the file in your own workspace. And the colleague promotes his changes to the common backing stream before you do.

Before you can promote your own changes to the backing stream, you must first *merge* (that is, combine) the changes from your colleague's version with your changes. Merging ensures that no one's work is inadvertently lost or overwritten in a concurrent development environment.

The File Browser makes it easy to tell which files require a merge prior to promotion. If a file needs to be merged in your workspace, the Details pane shows the file as having **(overlap)** status -- that is, the work of one or more of your colleagues has overlapped your own work. If you select **Overlap** in the File Browser's Searches pane, the Details pane displays all your **(overlap)**-status files: all files in the depot that you must merge before you can promote them from your workspace to its backing stream.


## Higher-Level Merges

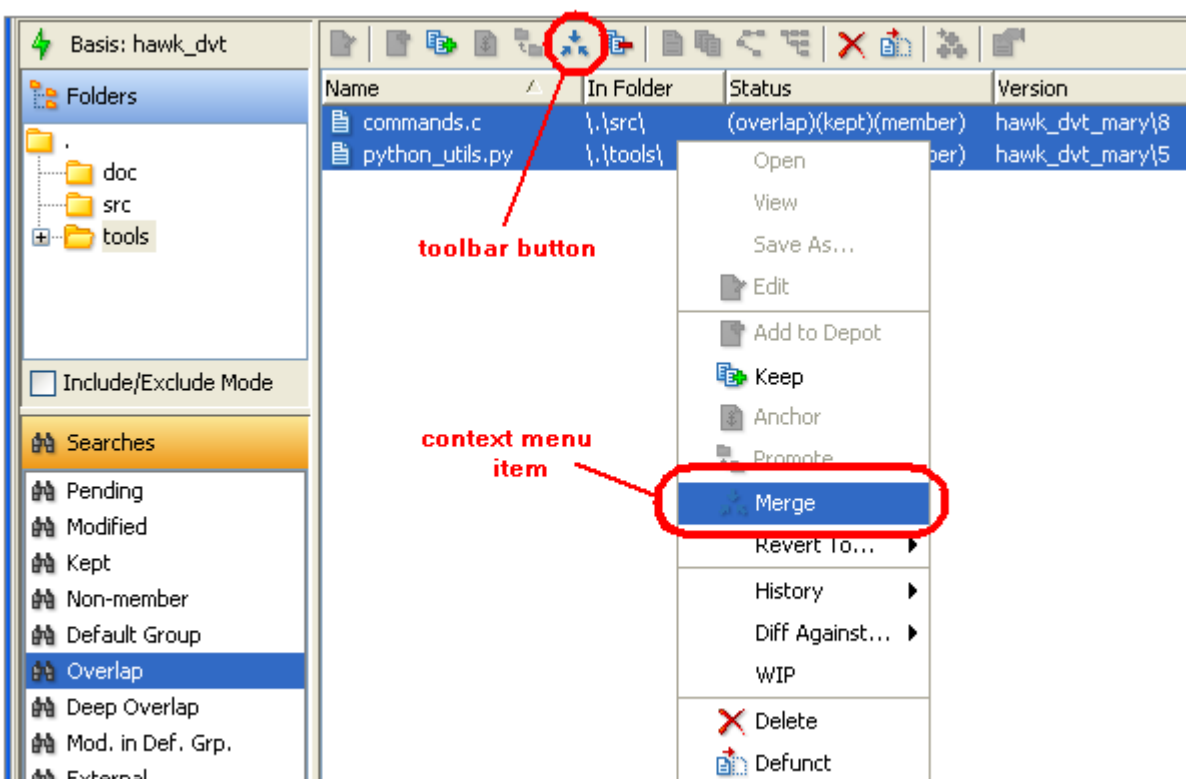
Although the above scenario is by far the most common one, AccuRev's flexibility allows for other merge scenarios, too:

- You can modify the file in your workspace by "pulling in" the changes from *any* version of the file, not just the backing stream's version.
- You can merge the versions of the file located in any two dynamic streams or snapshots. This involves using the *Change Palette*.

In all cases, however, you use the Merge tool in the same way, to combine the contents of two versions of the same file element.

## Invoking the Merge Command

No matter which tool you've configured to merge versions, you invoke the *Merge* command in the same way: select one or more files with (overlap) status, then click the  *Merge* toolbar button. Alternatively, choose *Merge* or *Merge From* on the selection's context menu. For example:



You can merge versions in the following contexts:

- In the File Browser (when invoked from a workspace), to merge from the version in the parent stream.
- In the Version Browser (when invoked from a workspace), to merge from an arbitrary version.
- In the History Browser (when invoked from a workspace), to merge from the version in a particular transaction.

- In the Change Palette, to merge versions in higher-level streams. (You must choose a workspace to use for the merge process.)

AccuRev processes the files you specify one-by-one. For each file, it does one or both of the following

- Prompts you to resolve namespace-level conflicts between the file in your workspace and the other specified version. This involves making choices in one or more dialogs.
- Invokes AccuRev's Merge tool or the other tool you've configured, in order to merge the contents of the two versions. AccuRev's Merge tool opens in a separate tab within the AccuRev GUI window; any other tool opens in its own window.

*Notes:*

- Invoking the Merge command on multiple files

If you select multiple files, you can choose to have the Merge tool process as many of them automatically as possible.



An automatic merge is possible if there are no *conflicting changes* between the two versions being merged. For files in which there are conflicting changes, the Merge tool always opens an interactive Merge tab.

- File vs. Version

In all cases, one of the contributors to a merge operation is the file in your workspace. If you've just saved the file with the *Keep* command, the file in your workspace is identical to the most recent version in your workspace stream. But if you invoke *Merge* on a file that you've edited but not yet saved with *Keep*, AccuRev uses the file, not the latest workspace-stream version. *Keep*'ing before *Merge*'ing is an AccuRev best practice.

## The AccuRev Merge Tool

This topic describes AccuRev's own *3-way merge* tool. You can also configure the GUI to use another text-file-merge tool.

### Merge Tab Layout

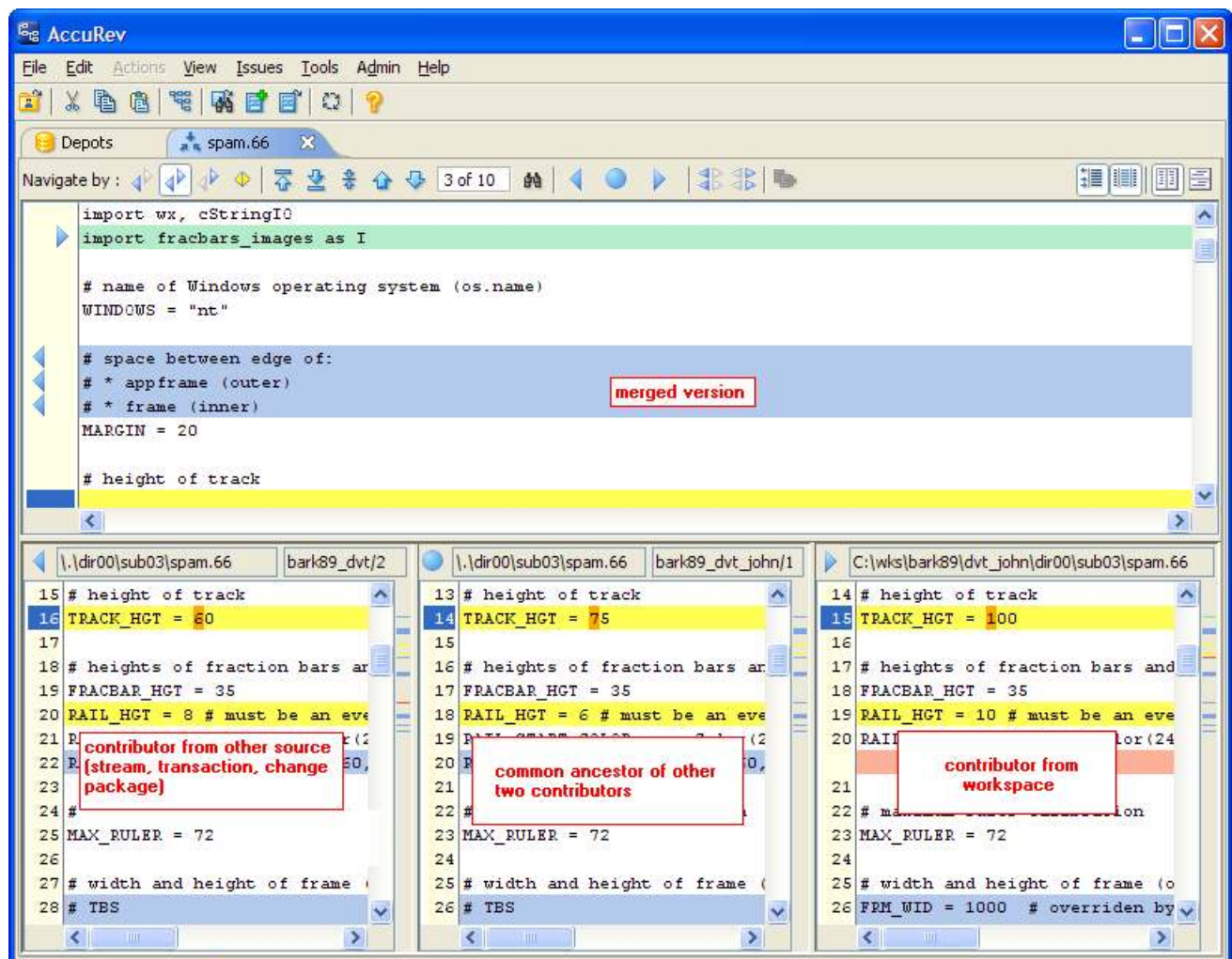
By default, the Merge tool displays the two contributors' contents in side-by-side panes with the contents of the contributors' *closest common ancestor* in a third pane between them.

Note: Using a toolbar button, you can arrange the contributors' panes vertically, or toggle the visibility of the pane containing the closest common ancestor.

The file in your workspace is always on the right. Above them, the Merge tool displays the merged version. The contributor panes and closest common ancestor pane are synchronized: scrolling any one of them causes the others to scroll, too.

Special cases:

- HTML files
- Binary files



## Color-Coding of the Contributor Versions

Like the Diff tool, the Merge tool partitions the *contributor* versions' contents, displaying unchanged sections with a white background and *change sections* with colored backgrounds. The color-coding scheme is a bit different, because the Merge tool has the more complex job of distinguishing *conflicting changes* from non-conflicting ones. In addition, the color-coding scheme depends on which navigation mode you're working in.

- For changes that are traversed in the current navigation mode, the Diff tool's color-coding scheme is used.
- Changes that navigating in the current mode skips over are colored *gray*.



- For a conflicting change, the blocks in all contributors are colored *yellow*, no matter what the navigation mode is.

Example 1: In *My changes* navigation mode, non-conflicting additions that come from your version are colored *green*; non-conflicting additions that come from the other contributor are colored *gray*. In *Their changes* navigation mode, the color-coding of non-conflicting additions would be reversed.




Example 2: In *Conflicts* navigation mode, all conflicting changes are colored *yellow*; all other change sections are colored *gray*.

## Color-Coding of the Merged Version

The merged version is displayed with color-coding, too. Each colored block is a location where a change has been incorporated from one of the contributors into the merged version. When it starts, the Merge tool performs as much merging as it can automatically, applying all of the non-conflicting changes to the merged version. The text is always colored *gray*. Annotations in the left margin indicate the origin of the change:

- left-hand (or upper) contributor:  annotation
- right-hand (or lower) contributor:  annotation

The Merge tool cannot resolve conflicts automatically, so the location of each conflict is initially indicated in the merged version by a blank *yellow* block. As you make selections to resolve the conflicts, the blank yellow blocks get filled in with the selected text and are annotated:

- blocks that you select from the left-hand (or upper) contributor:  annotation
- blocks that you select from the right-hand (or lower) contributor:  annotation
- blocks that you select from the closest common ancestor:  annotation

In all cases, the text of a resolved conflict is colored *blue*.

## Resolving Conflicts

You can think of the Merge tool as a special-purpose text editor: it combines the contents of two files, producing a third file as the result. Like any text editor, the Merge tool provides you with a set of editing and navigation functions; you have a lot of flexibility in choosing which functions to use, and in which order.

But the Merge tool does enforce this requirement: you must resolve all the conflicting changes between the contributor versions before you can complete a merge operation. Thus, Merge tool usage follows this basic pattern:

- Navigate to the various conflict sections.
- Resolve each conflict by selecting a section from the left-hand (or upper) contributor, the right-hand (or lower) contributor, or the closest common ancestor. You can also resolve the conflict by selecting the sections from more than one of these sources -- even from all three -- in any order.
- (optional) Add manual edits to the merged version.

- Save the results.

These operations (except for manual editing) are controlled by Merge toolbar buttons, as described in the following sections.

## Navigating the Display and the Change Sections

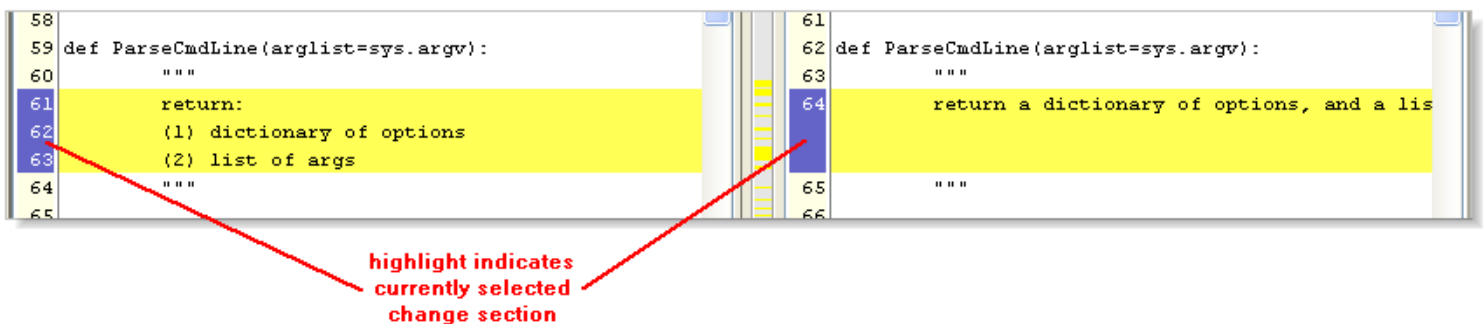
Some of the Merge tool's navigation facilities are the same as the Diff tool's:

- The panes have scroll bars, which are automatically synchronized with each other.
- The navigation buttons in the middle of the Merge toolbar work the same way as in the Diff tool, with a slight variation:



- If the *Their changes* mode button is selected, the navigation buttons traverse just the change sections in which the left-hand (or upper) contributor differs from the *closest common ancestor*.
- If the *All the changes* mode button is selected, the navigation buttons traverse all the change sections.
- If the *My changes* mode button is selected, the navigation buttons traverse just the change sections in which the right-hand (or lower) contributor differs from the *closest common ancestor*.
- If the *Conflicts* mode button is selected, the navigation buttons traverse just the sections with *conflicting changes*.

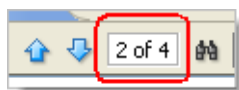
At any given time, one of the change sections is currently selected. The line numbers in both contributor panes are highlighted to indicate the selection.



Whenever you use one of these buttons (except for *Search*) to jump to a particular change section, the Merge tool remembers it as the *current change*. Selecting a particular change section by left-clicking it in either contributor pane makes it the current change.

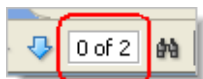
## The Change Section Counter

As you navigate among the change sections, a counter in the Merge toolbar tracks your location, taking into account which navigation mode is selected.



For example, "2 of 4" might mean "you are currently in *My changes* mode, and you are currently at the 2nd of 4 such changes".


The counter typically changes as you switch navigation modes.

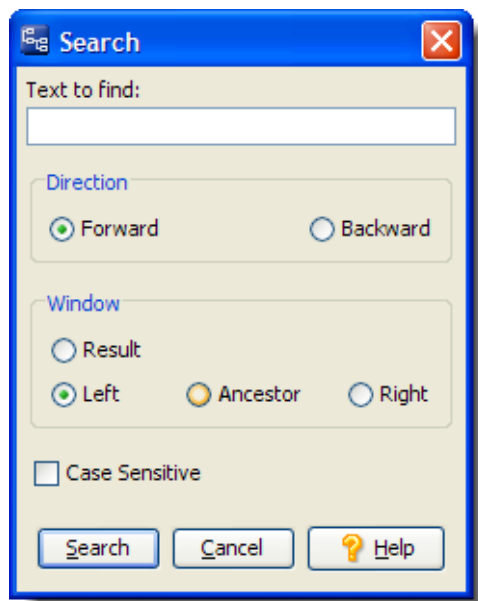


In some cases, the first number may be "0". This occurs when:

- You've just switched modes, and the current change section is not selected by the new mode.
- In Conflicts mode, you've just resolved a conflict, by selecting a section from one contributor. The current change section is no longer a conflict, so it's no longer selected by your current navigation mode.

## Searching for Text

In addition to navigating among the change/conflict sections, you can search for any text string in any pane, using the  *Search* toolbar button (keyboard shortcut: **Ctrl-S**).





You can also invoke this command from the context menu that appears when you right-click anywhere in either contributor pane.




## Selecting a Section from a Contributor (or Multiple Contributors)

Sometimes, the Merge tool can construct a merged version without any help from you. This occurs if all the change sections are non-conflicting. The Merge tool just applies all the changes to the merged version and announces that it's done. This section describes the more interesting case: the contributors have one or more conflicts, which *you* must resolve.

The Merge tool automatically jumps to the first conflict. That is, it scrolls to the first yellow-highlighted change section in the contributors, and the corresponding blank yellow block in the merged version. It's up to you to decide which of the changes is to be incorporated into the merged version. You indicate your decision using the conflict-resolution toolbar buttons.



- Click the  *Take common ancestor change* toolbar button to incorporate the original block from the closest common ancestor.
- Click the  *Take my change* toolbar button to incorporate the changed block from the right-hand (or lower) contributor (file in your workspace).
- You can take two of the blocks or all three of them, in any order:

1. Click the  o  r  or conflict-resolution button to select one block.





2. Hold down the **Ctrl** key, and click another conflict-resolution button. The newly selected block is sent to the merged version, below the previously selected block. The conflict-resolution buttons are annotated with numbers, indicating the order you selected. (And the tooltip reminds you of this "multiselect" capability.)

3. Optionally, repeat the preceding step to include the third block, as well.


You can undo a multiselect by clicking any of the conflict-resolution buttons *without* holding down the **Ctrl** key. Typically, you'll need to do some manual editing to "smooth out" the combined changes of a multiselect.

Note: You cannot undo a multiselect once you've performed a manual edit.]

The following two conflict-resolution buttons make large-scale changes -- use them carefully! They both effectively override the Merge tool's automatic inclusion of non-conflicting changes into the merged version.



- The  *Revert all of my changes* button makes the merged version the same as the left-hand (or upper) contributor version.
- The  *Use only my changes* button makes the merged version the same as the right-hand (or lower) contributor version.

When you click one or more conflict-resolution buttons to resolve a conflict, the selected block(s) appear in the merged version and the highlight changes from yellow (unresolved) to blue (resolved). This also decrements the change section counter in the Merge toolbar.

Use the  *Next* navigation button to scroll down to the next conflict. (It may take more than one click if you are not in *Conflicts* navigation mode.) Again, use one of the "take change" buttons to select the text from the left contributor or the right contributor (or both of them).

You don't have to resolve the conflicts in order. You can jump around as much as you want among the change sections, or scroll through unchanged sections to look up information that affects your merge decisions.

## Changing Your Mind

You can change your merge decisions as much as you want. For example, you can navigate to a particular change section where you had selected  *Take their change*, and then click the  *Take my change* button. This swaps out their change and swaps in your change.

**Warning:** swaps of this kind also discard any manual edits that you have made in that section after initially selecting it.

If you want to discard *all* your merge work, close the Merge tab to abandon the merge session entirely.



A confirmation dialog box appears, making it less likely that you'll discard your merge work accidentally.


## Manual Editing

At any time during a Merge tool session, you can manually edit the contents of the merged version. Just click anywhere in the pane containing the merged version, and type. The **Delete** and **Backspace** keys work as expected. Using context (right-click) menus or the GUI main menu, you can *Cut*, *Copy*, and *Paste* sections of text that you've highlighted with the mouse. You can also use the common keyboard shortcuts: **Ctrl-C** or **Ctrl-Ins** to copy, **Ctrl-V** or **Shift-Ins** to paste.

**Warning:** If you make changes manually within a conflict section that you've resolved with one of the conflict-resolution buttons, the manual changes will be overwritten if you return to that section and click any conflict-resolution button.

## Joining Change Sections

Sometimes, there is a block of lines in a contributor version that you consider to be a single change section, but that the Merge tool decides are two separate sections. You can combine the two change sections:


1. Right-click in the contributor section, and choose *Join Sections* from the context menu.
2. The mouse pointer changes to . Click the adjacent section to be joined with the first section.

The Merge tool joins the sections, then readjusts its analysis of the contributors' change sections. Typically, this changes the number of conflict sections.

### Saving the Merged Version

When you resolve the last remaining conflict by clicking one of the conflict-resolution buttons, the Conflicts counter goes to "0 of 0" and the Merge tool displays this message window:



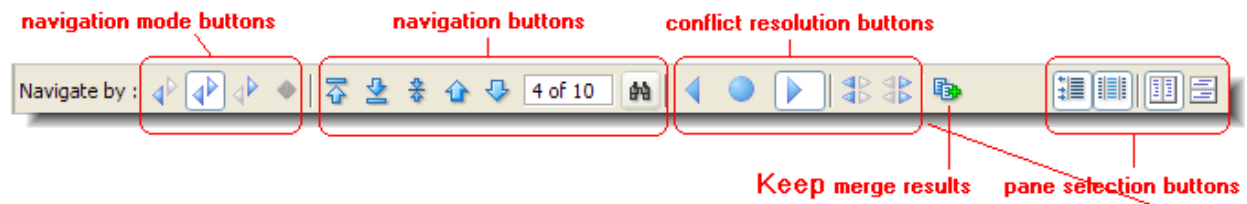
- Clicking *Keep & Exit* ends the merge session immediately, closing the Merge tab. AccuRev overwrites the file in your workspace with the merged version. Then, it *Keeps* a new version in your workspace stream, to preserve the merge results. The *overlap* status of the file is removed, so that you can now *Promote* the new version to the backing stream.
- Clicking *Review* continues the merge session. This enables you to review your work, change some of your merge decisions, and perform manual edits. At any time, you can click the  *Keep* button in the Merge toolbar to complete the merge process.

You can also cancel the entire merge during a review pass by closing the Merge tab without performing a *Keep*.

### Notes About Merge Behavior

As of AccuRev Release 4.9, the *merge* command is now smarter about simple merges, resulting in simplified display of merge lines in the Version Browser, and simplified change packages. For example, assume two developers are working in parallel on the same element, with a common ancestor, and there are only regular versions (that is, no patches, merges, or reversions) between the current versions and the common ancestor. When these are merged, the resulting version of the merge is rebased to point to the merged version as its basis. Also, the new basis version is added to the Comment field upon *keep*.

## Commands Available in a Merge Tab



### Navigation Mode Radio Buttons



#### ▢ Their changes

When this radio button is selected, the navigation buttons take you to the *change sections* in which the left-hand (or upper) contributor differs from the *closest common ancestor*.

#### ▢ All the changes

When this radio button is selected, the navigation buttons take you to *all* the *change sections*.

#### ▢ My changes

When this radio button is selected, the navigation buttons take you to the *change sections* in which the right-hand (or lower) contributor differs from the *closest common ancestor*.

#### ◆ Conflicts

When this radio button is selected, the navigation buttons take you to *all* the sections that contain *conflicting changes*.

### Navigation Buttons



#### ⬆ First change/conflict

Go to, and select, the first change section (or unresolved conflict section)

#### ⬇ Last change/conflict

Go to, and select, the last change section (or unresolved conflict section)

#### ⌄ Center current change/conflict

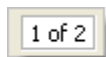
Scroll the contributor panes (if necessary), to make the currently selected change section visible.

#### ⬆ Previous change/conflict

Go to, and select, the preceding change section (or unresolved conflict section)

#### ⬇ Next change/conflict

Go to, and select, the next change section (or unresolved conflict section)

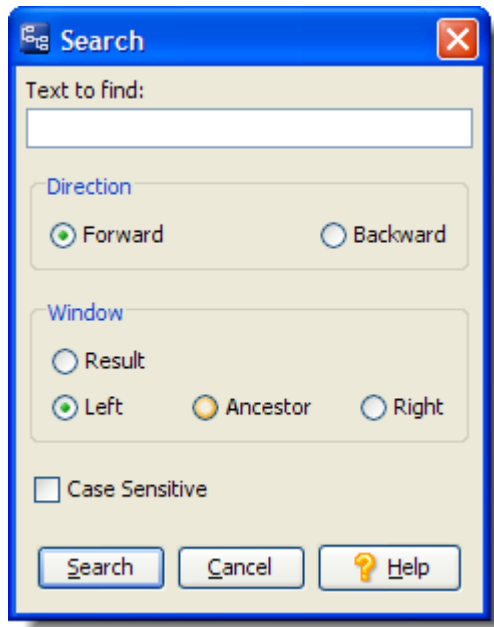


### Change section counter

Display only, not a button. See [The Change Section Counter](#) on page 235.

### Search

Bring up a dialog to define a search in one of the panes for a specified text string.



### Conflict Resolution Buttons



Invoking a "Take ... change" operation replaces a previously chosen change section at that location. It also wipes out any manual edits you may have performed in that change section.

#### Take their change

Copy the currently selected change section in the left (or upper) contributor pane to the results pane.

#### Take common ancestor

Copy the the original block from the closest common ancestor to the results pane.

#### Take my change

Copy the currently selected change section in the right (or lower) contributor pane to the results pane.

#### Revert all my changes

Make the merged version the same as the left-hand (or upper) contributor version.

### **Use only my changes**

Make the merged version the same as the right-hand (or lower) contributor version.

### **The "Keep merge results" Button**

#### **Keep results and close**

Preserve the current contents of the merge results pane as a new version of the element, using the *Keep* command. (The *Keep* dialog box does not appear.) The Merge tab for this file then closes automatically.

### **Pane Selection Buttons**



#### **Show results of edits**

Toggle the visibility of the merged version pane. This enables you to see more data in the two contributor panes.

#### **Show HTML result**

(appears for file with *.htm* or *.html* suffix only) Toggle switch: display the context of the merged version as plain text, or render the contents as HTML.

#### **Show Ancestor Panel**

Toggle switch: display/hide the closest common ancestor pane.

#### **Horizontal Layout**

(default) Places the "other" contributor pane to the left of the "workspace" contributor pane.


#### **Vertical Layout**

Places the "other" contributor pane above the "workspace" contributor pane.

## **The Search Command**

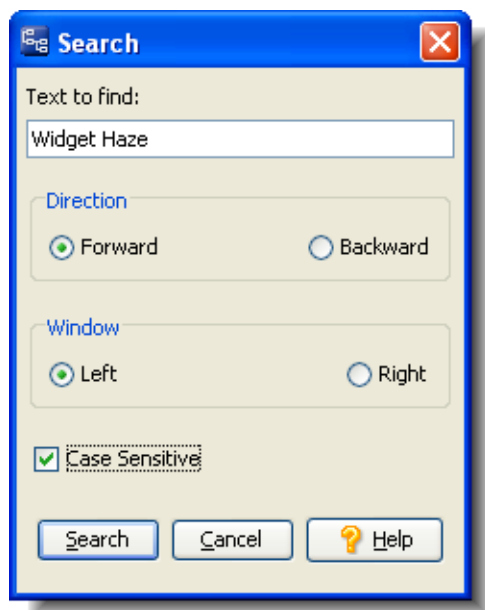
The *Search* command in the AccuRev Diff tool or Merge tool searches for a specified character string in one of the tools three panes -- workspace contributor (lower right), stream contributor (lower left), or results (top).

### **Invoking the Search Command**

Click the  button in the Merge toolbar.

### **Using the Search Dialog**

Type a character string in the text box, and make sure the *Direction*, *Window*, and *Case Sensitive* options are set in the desired way. Then click the *Search* button.



## The Merge, Patch, and Reverse Patch Algorithms

The AccuRev GUI uses the same tool to perform interactive *merge* operations, interactive *patch* operations, and interactive *reverse patch* operations (*Revert* command) on the contents of a text-file element. In all these operations:

- AccuRev analyzes two *contributor* versions of a file, along with a third version designated as the *closest common ancestor* of the two contributors.
- The two contributor versions' contents are combined to produce a new version of the file, which is saved in the repository with a *Keep* command. Sometimes, the new version can be produced completely automatically; other times, you have to interactively resolve *conflicts* between the contributor versions.

The only differences among the several operations are in which versions are designated to be the contributors and the closest common ancestor. These differences are detailed below.

### Notes:

- Using a third-party tool for merge, patch, and reverse patch operations  
By default, merge, patch, and reverse patch operations are performed by AccuRev's own Merge tool. If you configure a third-party text-file-merge tool, it will be used for all these operations. The algorithm used by the third-party tool is not necessarily the same as that used by AccuRev's Merge tool.
- Merging, patching, and reverting namespace changes  
Any merge, patch, or reverse patch operation can involve *namespace changes* in addition to (or instead of) *content changes*. AccuRev compares the pathnames of the two contributor versions, and in some cases applies a pathname change with the *Rename*

command. If both a *Rename* and a *Keep* are performed, the *Rename* transaction comes first.

See [Resolving Namespace Conflicts](#) on page 247 for more details.

## Merge: Incorporating Content Changes

In a *Merge* or *Merge From* command, the analysis performed on the two contributor versions goes beyond a simple *diff*: AccuRev determines how each *difference* represents a change from the *closest common ancestor* version. (To emphasize the *Merge* tool's perspective, we use the term “change section” to describe a location where the contributors differ from each other. In the context of the *Diff* tool, we use the term “difference section”.)

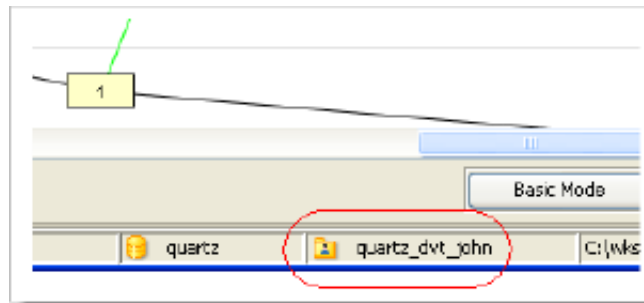
### Versions in a 3-Way Merge

The versions of an element that figure in AccuRev's **3-way merge** algorithm are:

#### 1. Workspace version (appears in lower right pane of Merge tool)

This is also called the “to” version. The merge results will be saved as a new version in some workspace, replacing the version displayed in this pane. Which workspace is the new version created in?

- In the most common merge scenario, you invoke the *Merge* command in a File Browser open on a particular workspace. The new version is created in that workspace.
- You can invoke *Merge From* or *Patch From* in any of the *version tools*.



- In this case, the new version is created in the workspace from which you launched the version tool. It's listed in the status bar at the bottom of the GUI window.
- If you invoke *Merge* from the Change Palette, you are prompted to specify a workspace in which to perform the merge. The new version is created in that workspace.

#### 2. Stream version (appears in lower left pane of Merge tool)

This is also called the “from” version.

- In the most common merge scenario, this is the version in the workspace's *backing stream*.
- If you're using one of the *version tools*, this is the version on which you invoked *Merge From* or *Patch From*.

·In the Change Palette, this is the version in the "source stream".

### 3. Closest common ancestor version (not displayed in Merge tool)

AccuRev takes into account previous merge operations, but not previous patch operations, in determining the closest common ancestor of the workspace version and the stream version.

Notes:

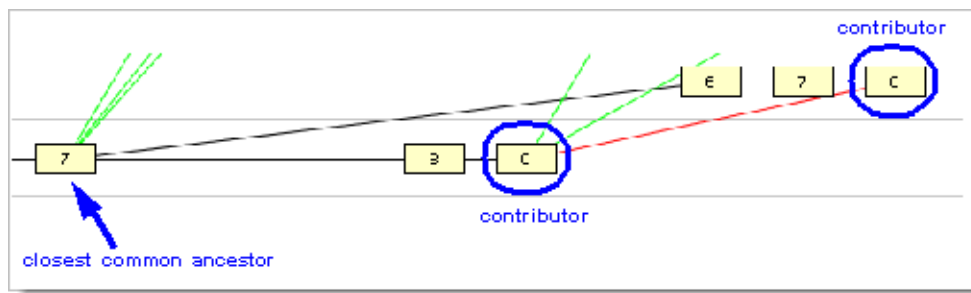
- Workspace File vs. Workspace Version

AccuRev always uses the file in the *workspace tree* as the first contributor. If you've just saved the file with the *Keep* command, the file in your workspace is identical to the most recent version in your *workspace stream*. But if you've edited the file without *Keep*'ing it, there's a difference.

In all cases, the version currently in the *workspace stream* is used in the determination of the *closest common ancestor* version.

- How can I tell which is the closest common ancestor?

In the Version Browser, you can tell by visual inspection. In the AccuRev CLI, use the command **anc -c**.



### Handling Change Sections

AccuRev processes each change section by comparing (1) the workspace version's content, (2) the stream version's content, the closest common ancestor's content:

- **Non-conflicting change:** Only one contributor — either the 'from' version or the 'to' version — changed the content in the section; the other contributor didn't make a change. AccuRev automatically includes the change in the merged version.
- **Conflicting change:** Both contributors changed the content in the section. AccuRev highlights the section in yellow; you must resolve the conflict by selecting one contributor's change to be included in the merged version.
- **Identical change:** Both contributors changed the content in the section *in exactly the same way*. AccuRev doesn't flag this section at all; the agreed-upon change is automatically included in the merged version.

#### Example 1: non-conflicting change

Suppose a change section consists of 13 lines that occur in contributor #2 but not in contributor #1. To determine what kind of change this represents, the Merge tool looks at the corresponding location in the closest common ancestor version:

- If those 13 lines exist in the ancestor version, the Merge tool concludes that a change was made in contributor #1 (the lines were deleted) but no change was made in contributor #2 (the lines are still there).
- If the 13 lines do not exist in the ancestor version, the Merge tool concludes that a change was made in contributor #2 (the lines were added) but no change was made in contributor #1 (nothing was added).

In both these cases, there was a change from the common ancestor in exactly one of the contributors. The Merge tool deems this a "non-conflicting change". It incorporates the change (be it an addition, a deletion, or a revision of existing text) into the merged version.

#### *Example 2: conflicting change*

Let's take another example. A one-line error message has a slightly different wording in the two contributors:

- Contributor #1:  

```
#define E_COLOR498 "No color with that name was found."
```
- Contributor #2:  

```
#define E_COLOR498 "Color name unknown."
```

The following line occurs at the corresponding location in the closest common ancestor version:

```
#define E_COLOR498 "Huh?"
```

In this situation, the Merge tool finds a change from the common ancestor in *both* contributors, not just one of them. This is a "conflicting change" (or more simply, a "conflict"). The Merge tool doesn't try to decide which contributor's change is better. It just makes it easy for *you* to make this decision when you perform the merge.

#### *Example 3: identical change*

It sometimes happens that both contributors have made *the same* change from the common ancestor version. For example, both contributors might have replaced this error message:

```
Huh?
```

with this one:

```
No such color
```

The Merge tool does not identify this as a difference section, because there's no difference between the two contributors. It silently incorporates the agreed-upon change into the merged version.

## **Patch: Incorporating Content Changes**

In a *Patch From* command, the two contributor versions are the same as in a *Merge* command:

- The lower right pane contains the version in your workspace.
- The lower left pane contains the stream (or "from") version -- the version, located in some other stream, that you selected when invoking the *Patch From* command.

But the patch algorithm does not use the actual closest common ancestor of these two contributors as the third version. Instead, it regards the stream version as being the *head version* of a patch, and uses the corresponding *basis version* as the closest common ancestor. (See [Structure of a Patch](#) on page 172.)

Using the basis version instead of the actual closest common ancestor effectively modifies the algorithm for handling change sections, enabling AccuRev to distinguish changes that are "in the patch" from changes made in other versions:

- If a change in the "from" version is in the patch, patching works in exactly the same way as merging. This means that it is possible (and perhaps automatic) that the "from" version's change will be included in the results of the Patch From command.
- If a change in the "from" version is not in the patch, it is ignored -- the workspace version's change is automatically included in the results. (Why automatically? This case fits the definition of a non-conflicting change: "not in the patch" means that the text in the stream version is the same as the text in the basis version (playing the role of closest common ancestor). Since only one contributor, the workspace version, has a change that differs from the version designated as the closest common ancestor, that contributor's change is accepted automatically.)

## Reverse Patch: Removing Content Changes

In a *Revert* command, the changes in a specified set of versions are removed from a stream's current versions of those elements:

- When reverting a **transaction**, AccuRev regards each version in that transaction as being the head version of a patch, and removes all the changes in that patch from the element.
- When reverting a **change package**, AccuRev removes all the changes in each element's change package entry.

For more information on the structure of a patch and a change package entry, see [Patches and Change Packages](#) on page 172.

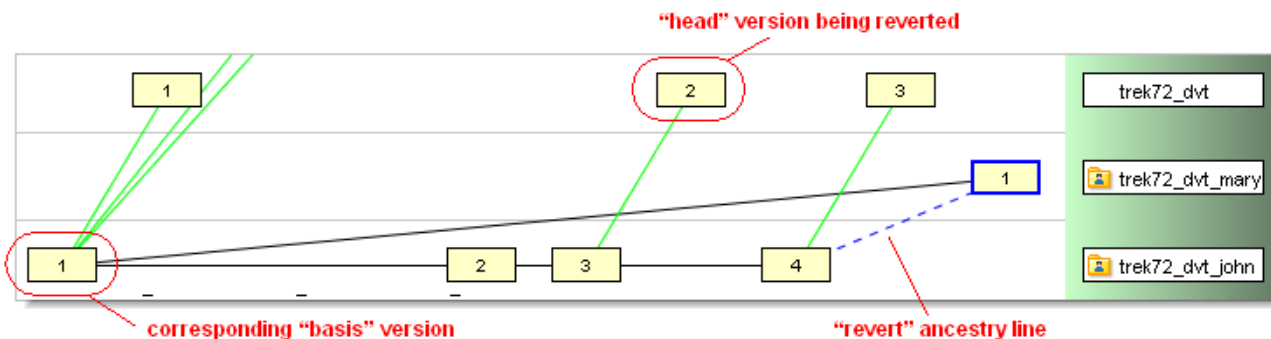
For each element it processes, the *Revert* command uses the AccuRev Merge tool (or user-configured tool) to perform the operation that removes a set of changes from the current version. We use the term **reverse patch** to describe this process, but it's really just another instance of effectively modifying the merge algorithm by switching around the versions.

When the Merge tool is invoked by *Revert*:

- The version you selected when invoking the Revert command is designated to be the closest common ancestor.
- The lower left pane contains the version currently in the stream.
- The lower right pane contains the version that was in the stream just before the version being reverted was promoted there.

In this algorithm's switching around of the versions, the basis version of the change package entry being reverted (or the basis version corresponding to the *Promote*'d version being reverted)

becomes the *direct ancestor* of the newly created version. The reverted element's *Version Browser* display shows this relationship:



## Resolving Namespace Conflicts

*Note:* This topic discusses conflicts that can occur when two developers change the pathname of an element. The same kind of conflicts can occur when two developers change the target element an *element link* or the target pathname of a *symbolic link*. The same dialogs appear in both kinds of conflict situations.

### Kinds of Namespace Changes

AccuRev distinguishes between these two kinds of namespace changes to an element (file, directory, or link):

- Rename: Changing the element's simple name (or "leaf name"):

`cmds.java --> commands.java`

- Move: Relocating the element to another directory in the depot::

`cmd_intf/commands.java --> cmd_intf/utils/commands.java`

*Note:* What about renaming the directory in which a file resides?

Renaming of the parent directory is not the same as moving a file to another directory. AccuRev considers this renaming:

`cmd_intf --> cmd_interface`

... to be change to the parent directory element, but not a change to the file elements within it. (This change to the parent directory has the *side-effect* of changing the pathnames of all elements below it.)

In a parallel development environment, namespace changes can *overlap* in the same way as content changes. For example, you and a colleague might rename the same file in your workspaces:

`cmds.java --> commands.java`

`cmds.java --> all_commands.java`

If your colleague promotes his change first, then the file will get **(overlap)** status in your workspace. Before you can promote the file, you must use the *Merge* command to resolve the namespace conflict.

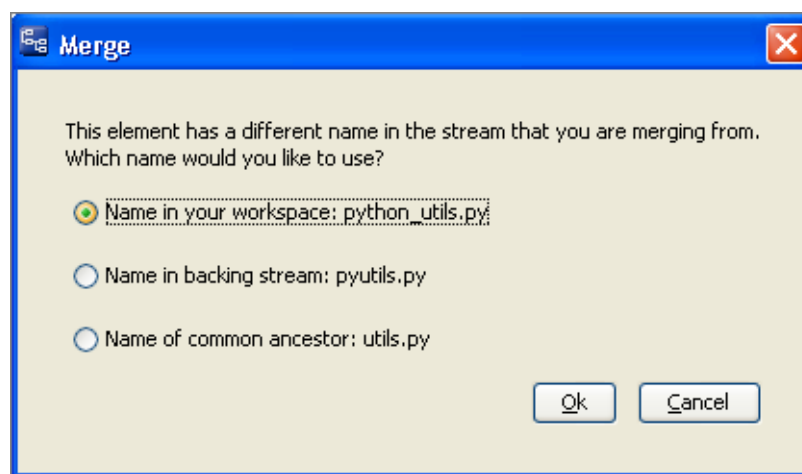
In general, your work on a file element can overlap with a colleague's work in three ways:

- You both have made a namespace change of the **rename** kind.
- You both have made a namespace change of the **move** kind.
- You both have made a content change to the file.

Any combination of these kinds of changes is possible. AccuRev first prompts you to resolve the namespace conflicts, if any, as described below. Then, it passes control to the Merge tool (or another tool that you've configured), to perform the content merge.

## Resolving a "Rename" Conflict

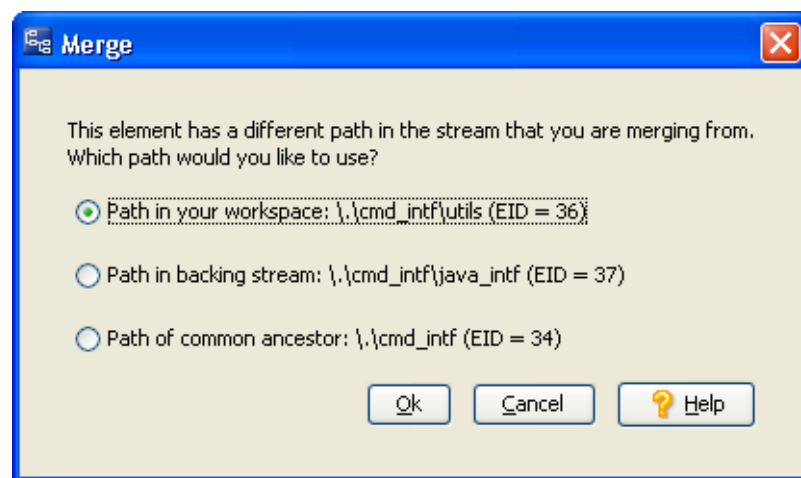
A namespace conflict of the **rename** kind occurs when two versions' simple names differ from the simple name of their *closest common ancestor* (and also differ from each other).



AccuRev allows you to resolve the conflict by taking any of the three simple names.

## Resolving a "Move" Conflict

A namespace conflict of the **move** kind occurs when two versions are located in different directories than their *closest common ancestor* (and also different directories from each other).



AccuRev allows you to resolve the conflict by choosing any of the three parent directories as the file's location.

**Note:** *What are the "EID" annotations?* AccuRev tracks directory elements by their immutable *element-IDs* (EIDs). It lists the parent directories' EIDs in this dialog to help you identify them; the directories might have different names/pathnames in different streams.

## Merging Versions of a Binary File

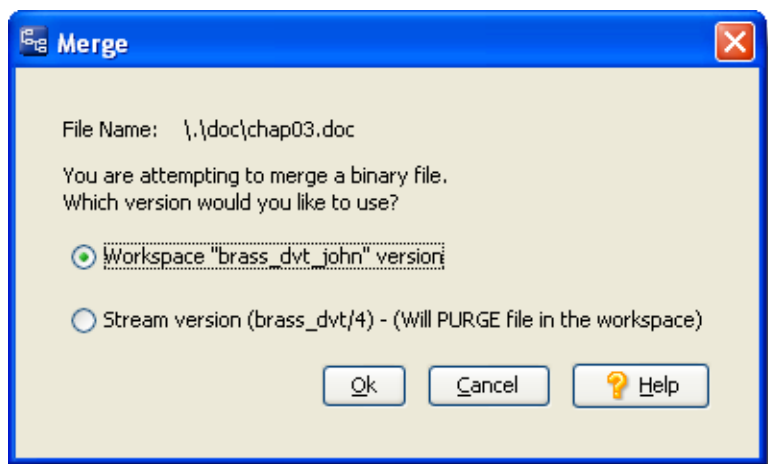
No generally accepted algorithm exists for merging the contents of binary-format files. But it is quite possible for a binary-format element to get into an *overlap* state in a concurrent development environment.

**Example:** Two team members might each revise the corporate logo file using an image editor, and then *Keep* it. One of them then *Promotes* the file to the backing stream, creating an overlap situation in the other workspace.

To resolve a binary-file overlap, you invoke the *Merge* command, just as for a text file. The Binary Merge dialog appears.

### Using the Binary Merge Dialog

The Merge tool, seeing that versions to be merged are in binary format, offers the only two possible choices:



#### Workspace version

Create a new version using the file in your *workspace tree*, and mark that version as having been merged with the version in the *backing stream*. You can now *Promote* this new version to the backing stream.

#### Stream version

This invokes the *Revert to Backed* command, which resolves the situation by purging the (**overlap**)-status version from your workspace. The workspace reverts to using the version of the file that was in the backing stream at the previous update.


The file's status becomes *(stale)*, because the version that previously caused the overlap is in the backing stream. To bring that version into your workspace, perform an *Update*.

## Merging Versions of an HTML File

AccuRev's merge algorithm treats HTML-format files just like all other text files: merging takes place line-by-line; no attempt is made to parse the files' HTML data structures in determining difference sections.

To help you determine whether the merge process is producing a valid (and desirable) result, an HTML viewer is built into the Merge tool.



At any time, you can click the  *Show HTML Result* button at the right side of the toolbar. (This button appears only when you're merging a file with a *.htm* or *.html* suffix.)

Clicking the *Show HTML Result* button displays a rendering of the merged HTML-format text in the upper pane. This button is a toggle: click it again when you're finished viewing the rendered HTML code and you want to return to performing the merge.

## Merging Versions of a Link

The *Merge* command can merge changes made to element in two different *contributor* versions.

### Basic Merge Algorithm

The basic procedure for handling changes in a merge operation is the same for *all* elements, including link elements:

- At the time of the merge, one contributor is located in a workspace, where the new version will be created); the other contributor is the version in the workspace's *backing stream* or some other *dynamic stream*. (You might have specified this second version as part of a particular *transaction* or *change package*.)
- If just one contributor makes a change from the contributors' *closest common ancestor*, AccuRev incorporates the change into the new, merged version automatically.
- If both contributors make a change from the contributors' closest common ancestor, AccuRev regards this as a *conflict*, which you must resolve during the merge process.

For a more detailed discussion, in the context of changes to the contents of a text file, see [Handling Change Sections](#) on page 244.

### What Constitutes a Change to a Link?

Given that the algorithm for handling changes is basically the same for all elements, the only thing special about merging versions of a link is the nature of changes to a link element.

- AccuRev tracks namespace changes to a link:
  - Initial creation of the link. Note: A link object is created as either an element link or a symbolic link. You cannot switch the type of an existing link object.

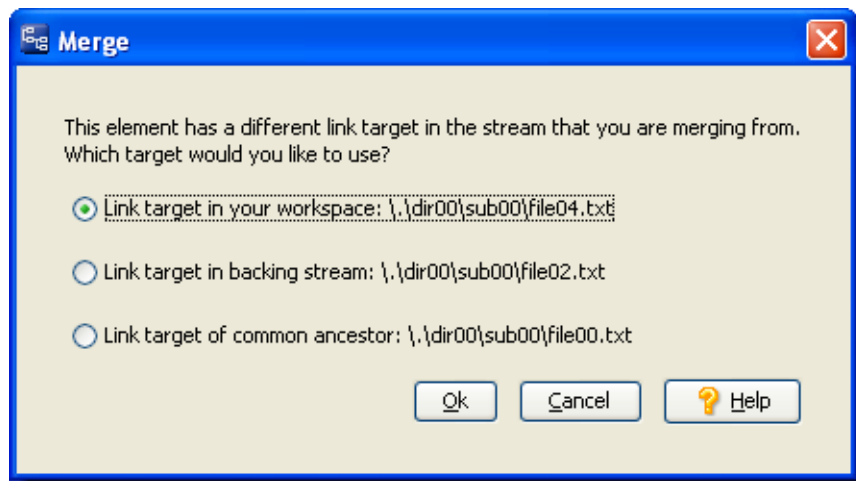
- Renaming the link.
- Moving the link to another location in the depot's directory tree.

Each of the above changes creates a new version of the link element.

- AccuRev also tracks changes to the contents of a link -- that is, changes to the link's target element (for an **element link**) or to the link's target pathname (for a **symbolic link**). Each such change creates a new version of the link element.

## Link Merge Procedure

Execution of the *Merge* command on a link element never involves the Merge tool (or third-party tool), which is designed to process the contents of text files.



A conflict at the content-level merge is handled by a link-specific dialog:

A conflict at the namespace-level merge involves making choices in the same one or two dialogs used for all kinds of elements.

## Merging Versions of a Directory

The *Merge* command can merge changes made to a **directory** (folder) element in two different **contributor** versions.

### Basic Merge Algorithm

The basic procedure for handling changes in a merge operation is the same for *all* elements, including directory elements:

- At the time of the merge, one contributor is located in a workspace, where the new version will be created; the other contributor is the version in the workspace's **backing stream** or some other **dynamic stream**. (You might have specified this second version as part of a particular **transaction** or **change package**.)
- If just one contributor makes a change from the contributors' **closest common ancestor**, AccuRev incorporates the change into the new, merged version automatically.

- If both contributors make a change from the contributors' closest common ancestor, AccuRev regards this as a *conflict*, which you must resolve during the merge process.

For a more detailed discussion, in the context of changes to the contents of a text file, see [Handling Change Sections](#) on page 244.

## What Constitutes a Change to a Directory?

Given that the algorithm for handling changes is basically the same for all elements, the only thing special about merging versions of a directory is the nature of changes to a directory element.

- AccuRev tracks only namespace changes to a directory:
  - Initial creation of the directory.
  - Renaming the directory.
  - Moving the directory to another location in the depot's directory tree.

Each of the above changes creates a new version of the directory element.

- AccuRev does not track changes to the contents of a directory. For example:
  - Adding a new element to a directory or renaming one of its elements not a change to the directory -- it's a change to that particular element.
  - Similarly, removing (with the *Defunct* command) an element from a directory is not a change to the directory -- it's a change to that particular element.

Each of the above changes creates a new version of the element within the directory, but does not affect the directory element itself.

## Directory Merge Procedure

Since AccuRev tracks only namespace changes to a directory, not content changes, execution of the Merge command on a directory element never involves the Merge tool (or third-party tool). A conflict at the namespace-level merge involves making choices in the same one or two dialogs used for all kinds of elements.

## The Merge Complete Dialog

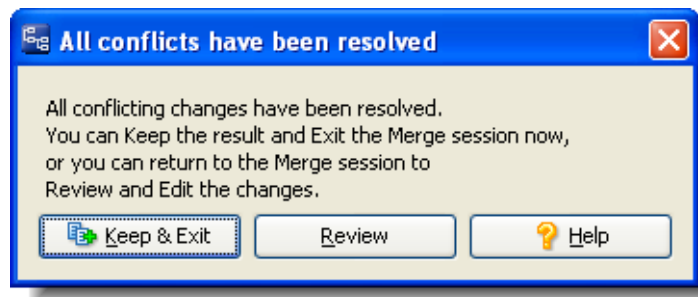
The Merge Complete ("All conflicts have been resolved") dialog appears during a *Merge tool* session, when you accept a change that decrements the Conflicts count to "0 of 0".

## Using the Merge Complete Dialog

You have these choices:

### Keep & Exit

Ends the Merge session, closing the Merge tab. AccuRev overwrites the workspace file involved in the Merge with the merged version



**Note:** You can think of the Merge tool as a fancy text-editor. It modifies the contents of a text file in the workspace.

Then, it preserves this file by *Keep*'ing a new version in the workspace stream. If the file had (**overlap**) status in the workspace, that status is removed.

**Note:** The workspace file has (**overlap**) status when you're merging your workspace version with the version in the workspace's backing stream. When you're merging a Change Palette entry, using a workspace based on the destination stream, the workspace file does not necessarily have (**overlap**) status.

You can now *Promote* the new version.

## Review

Resumes the Merge session, leaving the Merge tab open so that you can review your choices and changes. You can revise your merge choices and make manual edits. When you're done, click the *Keep results and close* button on the Merge toolbar:



## 7. Security



This topic presents an overview of AccuRev's security-related features. We discuss and compare the following topics, and provide pointers to more detailed documentation.

### Users and Groups

Each AccuRev user must have an AccuRev username (also called a "principal-name"). AccuRev maintains its own user registry in the repository; it is separate from the registry maintained by your machine's operating system (or the network).

Optionally, you can create AccuRev user groups, and add users to the groups as "members". Starting in Version 4.5, groups can be nested within one another; that is, a group can be a member of another group. AccuRev groups are independent of operating system groups.

AccuRev groups are used by the *Access Control List* (ACL) facility to grant or deny access to a particular resource for an entire set of users. They are used by the *Locks* facility to specify a set of users to which a stream lock does or does not apply.

### Username and Groupnames

Each AccuRev group has a user-defined name. Usernames and groupnames share a "namespace" in the AccuRev repository. This means that a user and a group cannot have the same name.

### User Authentication

With a few exceptions, a user cannot execute AccuRev commands unless he is authenticated as an AccuRev user. For authentication purposes, each registered AccuRev user has a username/password pair recorded in the registry. A user's password can be empty.

AccuRev automatically distinguishes two categories of users — those who have non-empty passwords and those whose passwords are empty. The keywords *authuser* and *anyuser*, respectively, are used by the ACL facility to designate these categories.

#### **The "AccuRev Login" User-Authentication Scheme**

Most AccuRev users are authenticated by explicitly logging in to the AccuRev Server. Using the GUI or the CLI, the user logs in by entering his username and password (possibly empty). This launches a user session, which is typically of limited duration (a few hours). When the session expires, the user must login again to continue using most AccuRev commands.

A successful login command creates an encrypted *session file* that records the user's AccuRev username and password, along with the IP address of the client machine. If an AccuRev client command can be executed only by an authorized user, the command automatically sends the information in the session file to the AccuRev Server process. Thus, the user doesn't need to repeatedly 'remind' the AccuRev Server who (and where) he is.

If a user is already logged in, and he successfully logs in again — as the same user or a different user — the existing session file is overwritten.

## The "Custom" User-Authentication Scheme

AccuRev also supports script-based authentication of AccuRev users. For example, customers with an LDAP authentication scheme in place at their organization could configure a script to authenticate AccuRev users via LDAP. See the "AccuRev Security Overview" chapter in the *AccuRev Administrator's Guide* for further details.

## Locks on Streams

Each stream can have one or more locks on it. A lock prevents a certain set of users from using the stream to create new versions of elements. That is, it prevents execution of the *Promote* command -- either promoting from the designated stream, or promoting to the designated stream, or promoting in either direction.

A lock can be absolute, applying to all users. Alternatively, you can specify that a lock applies to a particular AccuRev user, or to a particular AccuRev group. Conversely, you can specify that a lock applies to everyone *except* a particular AccuRev user, or to everyone *except* a particular AccuRev group.

Locks can also prevent reconfiguration of the contents of a stream with the include/exclude facility.

For more on locks, see [The Locks Command](#) on page 53.

## Access Control List (ACL) Permissions

In addition to (or instead of) locks, each stream can have one or more permissions on it. Whereas a lock controls the ability to create new versions (through the *Promote* command), a permission is more general: in addition to controlling *Promote*, it controls the ability to read data from the stream, using such commands as *Annotate*, *Diff*, and *Open*. A permission also controls workspace-specific commands, such as *Update* and *Populate*.

Unlike locks, which always apply to individual streams, ACL permissions can be configured to apply to entire stream subhierarchies.

You can also create an ACL permission that applies to an entire depot. This provides a way of controlling access to all of a depot's file system data, in all streams. It also provides a way to control access to issues in a depot.

## Element ACLs (EACLs)

As of Release 5.2, AccuRev also provides element-level ACLs (or "EACLs"). EACLs can only be set by an AccuRev administrator, and can only be set and modified using CLI commands. GUI users can be denied access to elements or directories if an administrator has configured EACLs. Depending on the situation, you will either see an "Access Denied" error message, a (no such elem) status, or the element will be completely invisible to you, as if it doesn't exist:

- For commands that take element name(s), if you are denied access to the namespace or content, you are denied access.
- For commands that specify an eid, such as *cat*, the command will succeed if you are not denied access to that element. If you are denied access to the namespace of the element, the eid is displayed instead of the name.

- For commands that display transactions, such as `hist`, if you are denied access to the element in the transaction, the transaction is displayed, but “Access Denied” is displayed for the path name.
- For commands that take a transaction or issue, such as `promote`, if you are denied access to an element or elements in the trans or issue, the command succeeds and the element(s) are promoted, but the name(s) are not displayed.
- Commands that modify workspaces, such as `update` and `pop`, will succeed even if you are denied access to some of the elements, but those elements will not be brought down to the workspace. The exception to this is if you specify the element name, such as with `pop` or `co`. In this case the command will fail as if the element doesn’t exist.
- For commands such as `stat`, `dirstruct`, and `files`, where you specify an option or nothing instead of an element name, if you are denied access to either the namespace or content, nothing will be displayed for that element. For example, `stat -d` displays all the elements in the default group of the stream. If you are denied access to an element in the default group, it will not be displayed.
- When you are denied access to an element, AccuRev behaves as if the element doesn’t exist. Depending on the specific circumstances, AccuRev will either:
  - Filter out the denied element altogether, so that there is no indication that it even exists. this is the default behavior.
  - Return an Access Denied status. This occurs when the only element in a display is denied, and a blank display would be confusing (for example, if the Show Active Files display in the StreamBrowser contains only denied elements).
  - Display a (no such element) status. If your access to an element or namespace changes during a session, and you are now denied access to something that was already displayed, that element or folder will appear with the (no such element) status.
- If you have multiple Access Denied elements in the default group or in the Pending or Kept filters, the GUI will combine these into a single Access Denied row.

For information about element ACLs, see [Chapter 10 Element Security](#) in the *AccuRev Administrator’s Guide*, and the command summary for `eacl` on page 104 of the *AccuRev CLI User’s Guide*

## ACL Permissions and Time Considerations

ACL permissions apply to a stream regardless of any *basis time* on the stream. Similarly, ACL permissions can be placed on a *snapshot*, even though such permissions are necessarily created after the snapshot is created.

For more on ACL permissions, see the description of [The Security/ACL Subtab](#) on page 265.

## Restricting Access to Commands using Triggers

By default, any registered AccuRev user can execute any AccuRev command. Many organizations wish to restrict users' access to certain commands, such as the ability to maintain users, groups, and passwords, the ability to lock streams and create ACL permissions, and so on. Triggers provide a flexible mechanism for implementing command-based security.

Many AccuRev commands can be configured to "fire a trigger". This causes a user-defined script to execute ...

- either before the command executes (pre-operation trigger), or afterward (post-operation trigger)
- either on the client machine, or on the server machine

A pre-operation trigger can affect the execution of the command or cancel it altogether. Typically, a security-related trigger checks the identity of the user invoking the command, then decides whether or not to allow the command to proceed.

Some triggers are configured on a per-depot basis, using the *mktrig* command. These triggers monitor individual commands (*add*, *keep*, and *promote*). Three are pre-operation triggers that fire on the client machine; one is a post-operation trigger that fires on the server machine.

Other triggers are configured, on a per-depot or whole-repository basis, by placing a script in a well-known location on the server machine. These triggers monitor groups of commands, rather than individual commands.

For more on triggers, see [The Triggers Tab](#) on page 51, and also [Chapter 9 AccuRev Triggers](#) of the *AccuRev Administrator's Guide*.

## Which Security Feature Should I Use?

AccuRev's security features overlap to a considerable extent. For example, when a user invokes the *promote* command, any of these mechanisms can prevent the command from proceeding:

- a lock on the source or destination stream
- an ACL permission on the destination stream, on a higher-level stream, or on the entire depot
- a pre-promote-trig trigger script, which runs on the client machine
- a server\_preop\_trig trigger script, which runs on the server machine

How do you decide which feature to use in a given situation? There are no absolute rules, but here are some guidelines:

### To script or not to script?

The trigger mechanism depends on execution of user-supplied scripts, written in Perl, Python, or some other scripting language. There's a trade-off: scripting required development time and significant expertise, but is infinitely flexible.

In many situations, you may be able to use the AccuRev software distribution's sample Perl scripts, which are designed for fill-in-the-blanks customization.

It makes sense to implement as much security as possible with locks and ACL permissions (and perhaps the sample trigger scripts), before delving into original trigger scripting.

### Locks vs. ACL permissions

Roughly speaking, a lock controls the placing of data *into* a stream, whereas an ACL permission controls the reading of data *from* a stream. (There are plenty of exceptions to this general rule.)

Both locks and permissions can be targeted at specific users or groups. The ACL is more flexible: you can create any number of permissions for the same stream, but only limited number of locks.

### Running trigger scripts: client machine vs. server machine

Running trigger scripts on the client machine conserves networking and server resources. But keep in mind that all client machines must have copies of the scripts (or must have network access to a central script repository).

Running trigger scripts on the server machine provides administrative simplicity and centralized logging.

## File Locking

Parallel development is flexible and powerful, but it is not appropriate for every situation. Some organizations don't like the extra steps involved in merging, even though merging is largely automated. Some files cannot be merged, because they are in binary format. (The merge algorithm handles text files only, not binary files such as bitmap images and office-automation documents.)

### Serial Development and Parallel Development

Accordingly, AccuRev supports both *serial development*, through its **exclusive file locking** feature, and *parallel development*. Exclusive file locking can be implemented at the depot level, the workspace level, or the element level (when you place the file under version control or when you create a new version).

The serial development model places more restrictions on users in the edit stage, but it eliminates the merge stage altogether.

#### Example:

Here's a standard scenario, in which all the workspaces are in serial-development mode:

1. A user starts working on a file by specifying it in a Send to Workspace ("checkout") or Anchor command. The file changes from being read-only to writable.
2. AccuRev places an exclusive file lock on the file. This prevents the file from being processed with Send to Workspace, Anchor, or Keep in other workspaces.
3. The user can edit and Keep any number of private versions of the file in his workspace. Then, the user Promote's his most recently kept version to the backing stream. The exclusive file lock guarantees that no Merge will be required before this promotion.
4. After Promote records the new version in the backing stream, things return to the initial state: AccuRev releases the exclusive file lock, and the file returns to read-only status in the user's workspace.
5. A user in any workspace can now Send to Workspace or Anchor the file, which starts the exclusive-file-locking cycle again.

If exclusive file locking applies to a file element, and the element is not currently active in any sibling workspace:

- It is maintained in a read-only state in your workspace tree.
- If the element is **active** in a sibling workspace, you cannot make the element active in your workspace.
- If the element is not active in a sibling workspace, you can make it active in your workspace with the Send to Workspace or Anchor command. This makes the file in your workspace tree

writable. (You must do this before you can use Keep to create a new version.) You can also make an element active with Revert, Rename, Defunct, Paste Link, or the CLI command undefunct.

Note: In this context, workspaces are considered siblings if they promote to the same stream. If the stream hierarchy includes pass-through streams, workspaces can be siblings even if they have different parents.

An exclusive file lock on a file element is released when active development on the file ends in that workspace:

- A promote command sends your private changes to an element from your workspace stream to the backing stream.
- A purge command discards your private changes to an element.

Either way, the workspace returns to using a backing-stream version of the element.

## The Limited Effect of an Exclusive File Lock

Exclusive file locking does not freeze an element completely:

- The lock applies only within the scope of a particular backing stream. It doesn't affect other backing streams and the workspaces based on them.
- The lock acquired through workspace-level or depot-level exclusive file locking applies only to workspaces in serial-development mode. Users in parallel-development-mode workspaces can make changes and promote the changes to the backing stream.
- A lock placed on an individual file element in a workspace applies to all sibling workspaces, but not to other workspaces.
- The lock doesn't prevent the current version in the backing stream from being promoted to higher-level streams.

Exclusive file locking does not prevent any user from modifying any file with a text editor or IDE. AccuRev encourages users in serial-development-mode workspaces to "ask permission first": it maintains files in a read-only state, and makes a file writable when a user executes a *co* or *anchor* command on it. But users can modify a file "without asking permission", by changing the access mode (UNIX: *chmod* command, Windows: *attrib* command or *Properties* window) and then editing it. Such "unauthorized" changes can't be sent to the AccuRev depot, though: the exclusive file lock disallows a *Send to Workspace*, *Anchor*, or *Keep*.

## Anchor-Required Workspaces

AccuRev also offers a less-restrictive variant of exclusive file locking. **Anchor-required** workspaces allow parallel development, with multiple users modifying the same file at the same time (in their own workspaces). In an anchor-required workspace, *all* elements are maintained in a read-only state when you are not actively working on them. Using such a workspace is similar to working with exclusive file locking, except that you are not constrained by elements' activity in sibling workspaces:

- You can make a file element active with the Send to Workspace or Anchor command. This makes the file writable. You must do this before you can use Keep to create a new version.
- You can also make an element active with Revert, Rename, Defunct, Paste Link, or the CLI command undefunct.

## The Security/Users Subtab

The Users subtab of the Security tab displays the repository's set of registered AccuRev *users*, and provides commands for maintaining the set of users. Various security (and non-security) features involve users:

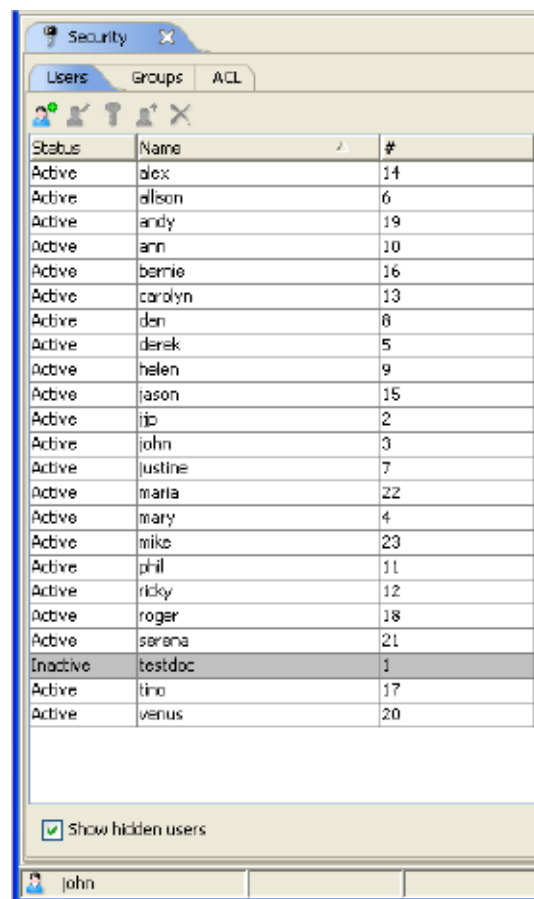
- Before executing any AccuRev commands, you must Login as a particular user.
- Access Control List (ACL) entries can apply to particular users or to particular groups.
- AccuWork issue records can contain fields whose value must be a particular AccuRev user.

### Opening a Security/Users Subtab

Choose *Admin > Security* from the GUI main menu to display the Security tab. Then, click the Users subtab.

### Security/Users Subtab Layout

The Users subtab contains a table with these columns:



Status	Name	#
Active	alex	14
Active	allison	6
Active	andy	19
Active	ann	10
Active	bernie	16
Active	carolyn	13
Active	den	8
Active	derek	5
Active	helen	9
Active	jason	15
Active	ijo	2
Active	john	3
Active	justine	7
Active	maria	22
Active	mary	4
Active	mike	23
Active	phil	11
Active	ricky	12
Active	roger	18
Active	serena	21
Inactive	testdoc	1
Active	tino	17
Active	venus	20

☒ Show hidden users

john

### Status

(appears if *Show hidden users* is checked) **Inactive** indicates that the username has been removed, and can't be used in a *Login* command, or in the definition of an *ACL* entry or a *lock*. A removed user can subsequently be *Reactivated*.

## Name

The username. You can change a username (for example, from **derek** to **dpost**) without disrupting AccuRev's bookkeeping, because it tracks user identities by their unique, immutable integer IDs.

A user cannot have the same name as any other user, or the same name as any group. See [User-Specified Names for AccuRev Entities](#) on page 14.

## #

The integer that uniquely identifies this user. This number cannot be changed. **Note:** The same "namespace" of integer IDs is shared by users and groups.

## Working in a Security/Users Subtab


You can use this subtab to activate/deactivate users, and to perform operations that maintain the user registry.

## Controlling Which Users are Displayed

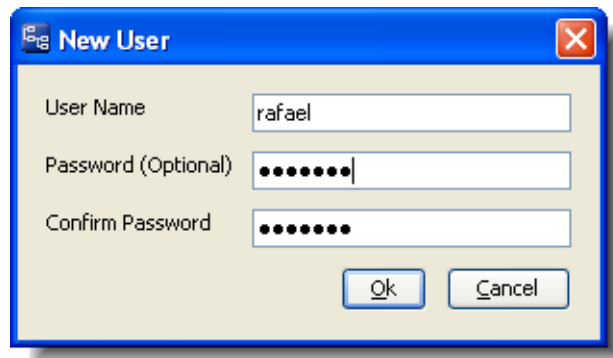
### Show hidden users

If checked, the table includes users that have been deactivated with the *Remove* command. This also adds a Status column.

## Creating a New User

Use the  toolbar button to create a new AccuRev user identity.

### Add User

A screenshot of a Windows-style dialog box titled "New User". It has a blue title bar with a close button (X) in the top right corner. The dialog contains three text input fields: "User Name" with the text "rafael", "Password (Optional)" with masked characters "••••••", and "Confirm Password" with masked characters "••••••". At the bottom right, there are two buttons: "Ok" and "Cancel".

Add a new user to the repository's user registry, optionally assigning a password to the user. In order to succeed, there must be at least one available user license.

## Operating on a Selected User

You can choose any of the following commands from the context menu of a selected user:

### Change User Name

Change the name that AccuRev associates with this user identity. See [User-Specified Names for AccuRev Entities](#) on page 14.

### Change Password

Change this user's password. **Note:** AccuRev allows you to change any user's password, not just your own. But your organization might use a trigger to disallow all but a few administrative users from changing another user's password.

See [User-Specified Names for AccuRev Entities](#) on page 14.

### **Reactivate User**

Reactivate a user who has been deactivated with *Remove User*. In order to succeed, there must be at least one available user license.

### **Remove User**

Deactivate the selected user. No one can Login as that user, but the user still appears in the repository's historical records, such as the History Browser. You can use the *Reactivate User* command to bring back a removed user.

## The Security/Groups Subtab

The Groups subtab of the Security tab displays the repository's set of user *groups*. Individual *users* can be members of any number of groups. Likewise, a group can belong to -- that is, be a subgroup of -- any number of other groups.

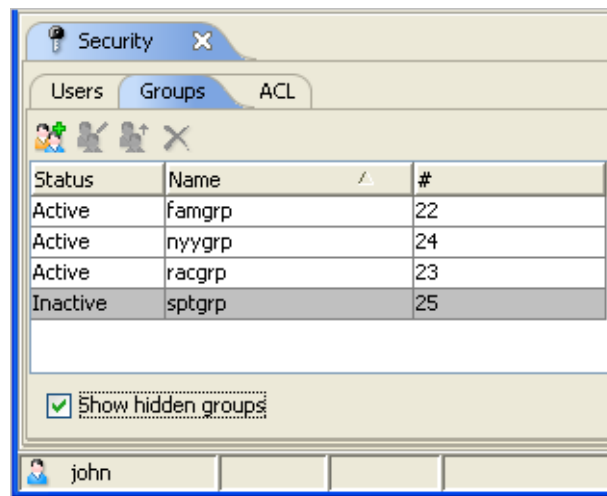
Access Control List (ACL) entries can apply to particular users or to particular groups.

### Opening a Security/Groups Subtab

Choose *Admin > Security* from the GUI main menu to display the Security tab. Then, click the Groups subtab.

### Security/Groups Subtab Layout

#### Status



(appears if *Show hidden groups* is checked) **Inactive** indicates that the group has been *Removed*, and can't be used in the definition of an *ACL* entry or a *lock*. A removed group can subsequently be *Reactivated*.

## Name

The group name. You can change a group name (for example, from **sptgrp** to **sport\_group**) without disrupting AccuRev's bookkeeping, because it tracks group identities by their unique, immutable integer IDs.

A group name can be up to 96 characters long. A group cannot have the same name as any other group, or the same name as any user. See [User-Specified Names for AccuRev Entities](#) on page 14.

## #

The integer that uniquely identifies this group. This number cannot be changed. **Note:** The same "namespace" of integer IDs is shared by users and groups.

## Working in a Security/Groups Subtab

You can use this subtab activate/deactivate groups, and to perform operations that maintain groups and their memberships.

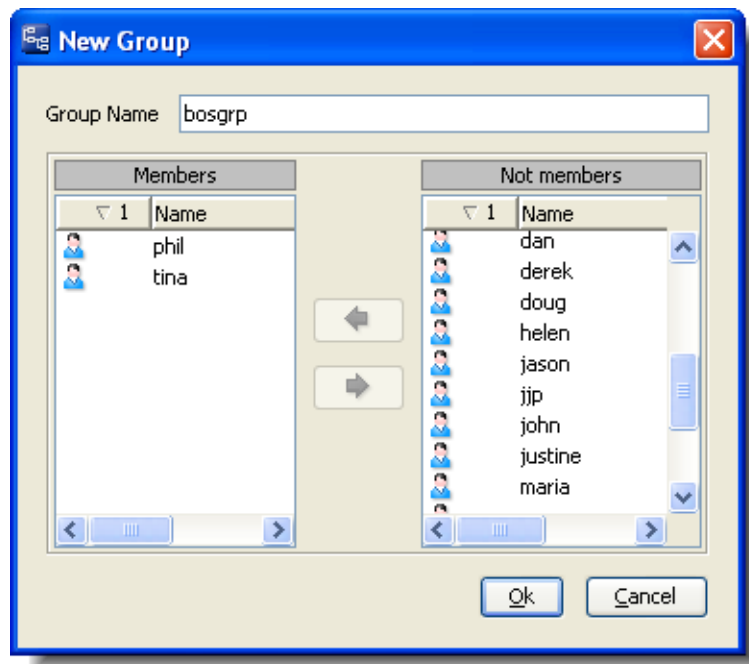
## Controlling Which Groups are Displayed

### ☒ Show hidden groups

If checked, the table includes group that have been deactivated with the *Remove* command. This also adds a Status column.



## Creating a New Group

Use the  toolbar button to create a new group.



### Add Group

A New Group dialog appears, in which you specify the group's name and its members.

To specify a group's membership, select one or more users and/or groups, and click the  or  button to move them between the **Members** and **Not members** lists.

A user can be a member of any number of groups. Likewise, a group can belong to (be a subgroup of) any number of groups.

#### Operating on a Selected Group

You can choose any of the following commands from the context menu of a selected group:

##### **Change Group**

Change the name that AccuRev associates with this group identity, or change the group's membership. A Change Group dialog appears, works the same way as the New Group dialog.

A user can be a member of any number of groups. Likewise, a group can belong to (be a subgroup of) any number of groups.

##### **Reactivate Group**

Reactivate a group that has been deactivated with *Remove Group*.

##### **Remove Group**

Deactivate the selected group.

## The Security/ACL Subtab

The ACL subtab of the Security tab displays the repository's set of access control list (ACL) entries, also called **permissions**. Each stream and depot in the repository can have any number of permissions. **Note:** For this command, "stream" can be a dynamic stream, a workspace stream, or a snapshot.

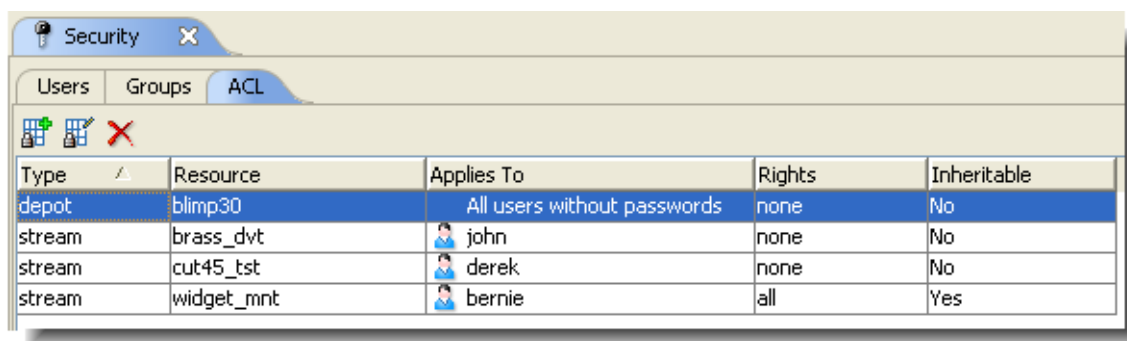
Each permission controls the ability of a particular user, or a particular set of users, to access that stream or depot.

### Opening a Security/ACL Subtab

Choose *Admin > Security* from the GUI main menu to display the Security tab. Then, click the ACL subtab.

### Security/ACL Subtab Layout

The table in the ACL subtab includes these columns:



Type	Resource	Applies To	Rights	Inheritable
depot	blimp30	All users without passwords	none	No
stream	brass_dvt	john	none	No
stream	cut45_tst	derek	none	No
stream	widget_mnt	bernie	all	Yes

## Type

The type of resource to which the permission applies: **Stream** or **Depot**.

## Resource

The name of the stream or depot to which the permission applies. The permission also applies to (is inherited by) the entire subhierarchy below that resource, but it can be overridden at a lower level.

**Note:** Because permissions are inherited down a depot's stream hierarchy, a permission on a depot is almost equivalent to one on the depot's base stream. The only difference is that a depot permission also affects access to the depot's issues.

## Applies To

The user or set of users to which the permission applies: all users with passwords (identified as "authuser" in the AccuRev CLI), all users without passwords ("anyuser" in the CLI), a particular user, or a particular group.

## Rights

**all:** the user(s) can access the resource.

**none:** the user(s) cannot access the resource.

## Inheritable

Depot permission:

**yes:** the permission applies to the depot, and also to the depot's entire stream hierarchy.

**no:** the permission applies only to the depot, not to any of the depot's version-controlled elements.

Stream permission:

**yes:** the permission applies to the entire subhierarchy below the stream in the Resource column.

**no:** the permission applies only to the stream in the Resource column, not to any lower-level stream.

## How Individual Commands Use the ACL

The following commands check ACL entries on one or more *dynamic streams*, *workspace streams*, or *snapshots* before proceeding. In the following, 'stream' can be a dynamic stream, workspace stream, or snapshot. If a version is being accessed from stream A, and that version is *cross-link*'ed to stream B, AccuRev checks the ACL permissions on stream A only, not on stream B.

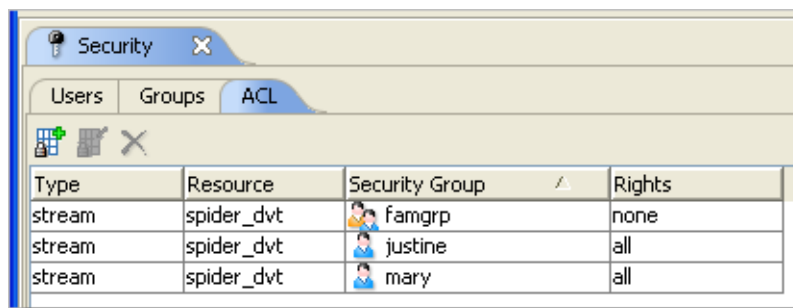
- *Anchor*, *Defunct*, *Populate*, *Revert to Backed*, *Revert to Most Recent Version*, *Update* and File Browser searches check the current workspace.
- *Annotate*, *View*, and *Send to Workspace* check the stream of the version being accessed. *Send to Workspace* also checks the current workspace.
- *Promote* checks the stream to which the version(s) are being promoted.
- *Include from Stream* and *Clear Rule* check both streams involved in the cross-link.
- The various *Diff Against ...* commands check the streams of both versions being compared.
- *Merge* checks the workspace/stream in which you're doing the merge, as well as any stream explicitly mentioned on the command line.
- *Remove* and *Reactivate stream* check the stream being changed.
- *New Workspace*, *New Stream*, and *New Snapshot* check the specified backing stream for the workspace/stream/snapshot being created.
- *Change Stream* checks the stream being changed (and, if appropriate, its new backing stream).
- *View Streams* checks the depot.
- In a *Stream Browser* tab, *Show History* checks the selected stream. In a *Depots* tab, *Show History* checks the selected depot.

## Conflicting Permissions

Two or more permissions on a resource can apply to the same user, or to the same stream. In such cases, an **all** permission overrides one or more **none** permissions. This makes it easy to implement "all but" access controls.

**Example 1:** "all but" at the user level

The permissions shown here prevent everyone in the **famgrp** group -- except for users **justine** and **mary** -- from accessing stream **spider\_dvt**.

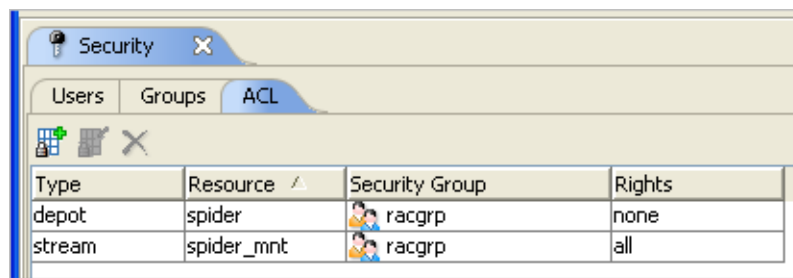


The screenshot shows a 'Security' dialog box with the 'ACL' tab selected. It displays a table of permissions for the resource 'spider\_dvt'.

Type	Resource	Security Group	Rights
stream	spider_dvt	famgrp	none
stream	spider_dvt	justine	all
stream	spider_dvt	mary	all

Example 2: "all but" at the stream level

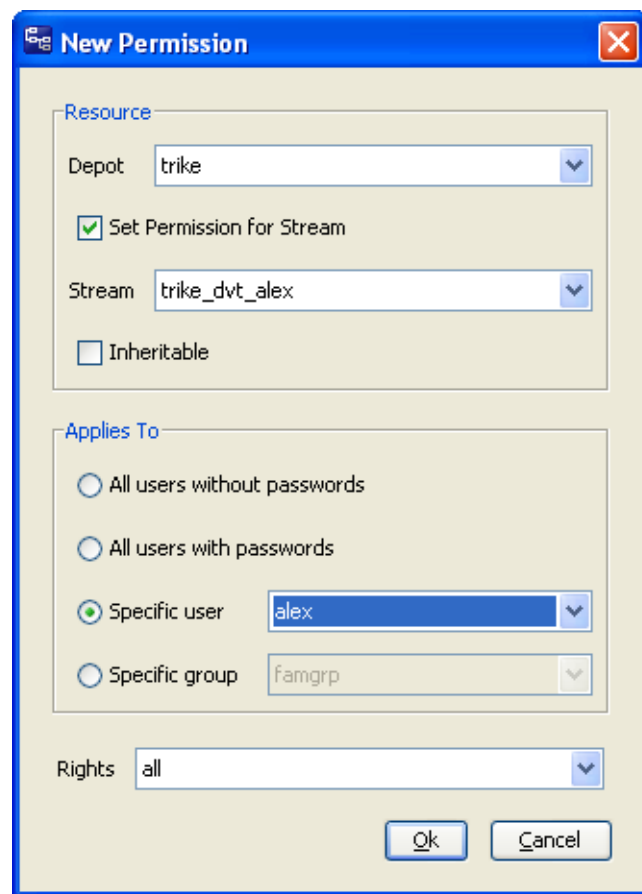
The permissions shown here restrict the **racgrp** group to working in the **spider\_mnt** stream subhierarchy. Members of this group cannot work with other streams in the **spider** depot.



Type	Resource	Security Group	Rights
depot	spider	racgrp	none
stream	spider_mnt	racgrp	all

## Working in a Security/Access-Control Subtab

Use the following commands to maintain the set of permissions in the repository's access control list.



**New Permission**

Resource

Depot: trike

☒ Set Permission for Stream

Stream: trike\_dvt\_alex

☐ Inheritable

Applies To

☐ All users without passwords

☐ All users with passwords

☒ Specific user: alex

☐ Specific group: famgrp

Rights: all

Ok Cancel

### Add ACL Entry

### Change ACL Entry

The same dialog enables you to create a new ACL permission, or to modify an existing one.

**Resource:** Select one of the repository's depots from the *Depot* listbox.

- If you want the permission to apply to the depot itself, leave the *Set Permission for Stream* checkbox cleared.
- If you want the permission to apply to one of the depot's stream, check the *Set Permission for Stream* checkbox, and choose a stream from the *Stream* listbox.
- If you want the permission to apply to the entire subhierarchy below the specified stream, check the *Inheritable* checkbox.

**Security Group:** Use the radio buttons (and listboxes) to specify an individual user or a single group to which the permission will apply. (You can't specify multiple users or groups, but you can produce the same effect by creating multiple permissions on the same resource.)

**Permission:** Select **all** or **none** from the listbox.

### Remove ACL Entry


Deletes the selected entry(s) from the access control list.

## The Available AccuRev Servers Dialog

(and the New Server and Edit Server dialogs)

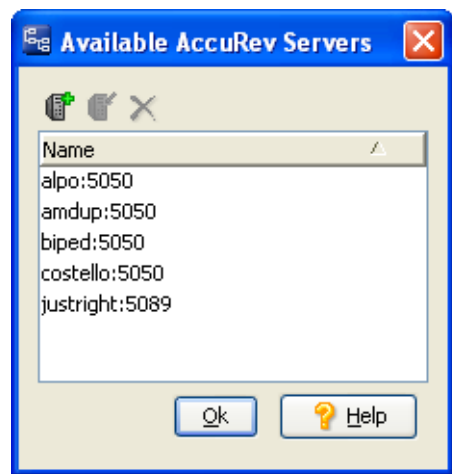
The Available AccuRev Servers dialog provides a way to maintain the list of AccuRev Servers to which you can *login*. Each *client* installation of AccuRev maintains its own AccuRev Server list, in file <AccuRev-install-dir>/bin/acclient.cnf.

### Invoking the Available AccuRev Servers Dialog

Choose *Tools > Login* from the GUI main menu. Then click the  button in the *Login* dialog.

### Using the Available AccuRev Servers Dialog

To switch to another AccuRev Server, select its entry, and click *Close*.



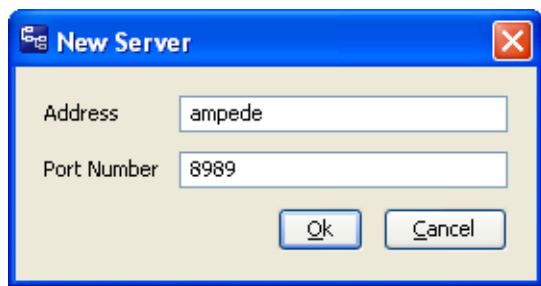
Click the Name column header to reverse the order in which the AccuRev Servers are listed.

## Toolbar Commands

The toolbar includes the following commands for maintaining the AccuRev Server list:

### Add Server

Identify an instance of the AccuRev Server, to which anyone using this client installation will be able to log in. A dialog appears, in which you identify the Server



The Address can be a name (e.g. *abbot* or *abbot.myfirm.com*) or an IP address (123.156.189.047). Be sure to specify the correct port number for that Server. No attempt is made to check the validity of the new entry.

### Edit Server

Modify the address and/or port number of the selected entry in the list. No attempt is made to check the validity of the modified entry.

### Remove Server

Remove the selected entry from the list. This does not affect that AccuRev Server process; it merely removes the Server from this client installation's list.

## A. Miscellaneous Error Displays

### No Query Results Error

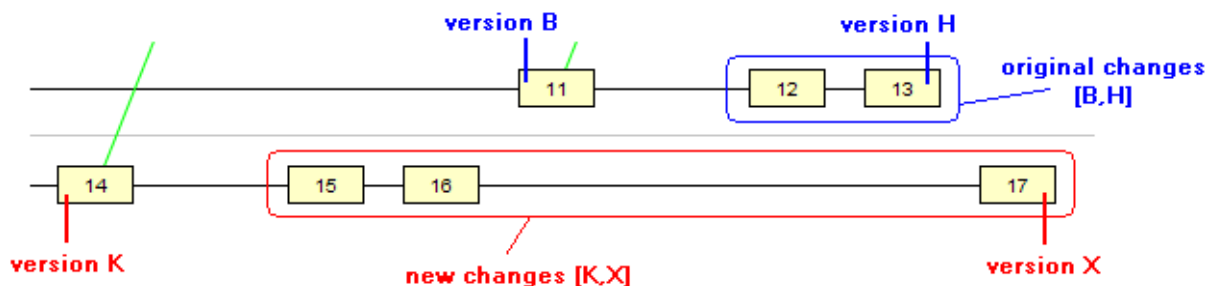
This error box appears when a set of AccuWork *issue records* is to be displayed, but no table display format has been defined in the Change Package Results pane of the Change Packages subtab of the Schema Editor tab.

### Diff Previous Transaction Error

This error box appears when the selected version is the *first* version created in a stream or workspace. Thus, the search for a prior version in the stream or workspace fails.

### The Issues Conflict Message Box

This error box appears when there is no way to produce a valid change package entry for an element, by combining your new set of changes with the existing change package entry. The requirement that the basis version must be an ancestor of the head version cannot be fulfilled.



You can remedy this situation by performing a *merge* at the element level. (There is no merge operation defined for change package entries.) In the example above, merging version H and version X would create a new version; a change package entry with the new version as its head can be combined with the existing entry.

### No Schema Error

This error box appears when AccuWork has not been configured in the current *depot*. Use the *Admin > Schema Editor* command to define a schema.

### Promote Coalesce Error

If a promote-by-issue operation (a standard child-to-parent promote, not a cross-promote) involves multiple issues whose change packages include the same element, AccuRev attempts to combine or *coalesce* those entries into a single, valid change package entry. If the element's

change package entries cannot be coalesced (caused, for example, by a “gap”), the promote operation fails.

If this occurs, please follow the steps described in “Cross-Promoting Issues to a Non-Parent Stream — Patch Required” in the *AccuRev Technical Notes*.

## Which update topic?

Which help topic would you like?

- General help on the Update command: see [Updating a Workspace](#) on page 28
- Help on the Update progress box only: see [The Update Progress Box](#) on page 133

Note: Why am I here? I was using the include/exclude facility or the Populate command.

- The commands that you execute in the File Browser's include/exclude mode complete their work by performing an Update of the workspace.
- The Populate command does not perform an Update, but these two commands use the same low-level routines to copy files from the repository to the workspace tree.

## B. AccuWork



### Working with Issue Records

Note: As of AccuRev 5.2, the AccuRev Java client will launch the AccuRev Web User Interface (Web UI) whenever you access an AccuWork issue under certain conditions. These conditions are listed in [Integration with the AccuRev Web UI](#) on page 7.

If AccuWork issues are viewed with the AccuRev WebUI at your site, please see the AccuRev Web User Interface help for more information.


If AccuWork issues are viewed with the AccuRev Java client, use the information in this chapter.

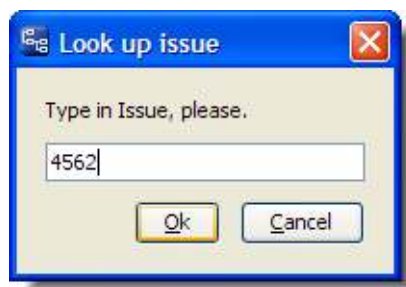
### The Edit Form Tab

The AccuWork Edit Form tab provides a familiar, easy-to-use interface for creating new *issue records* and modifying the contents of existing issue records. The edit form also provides access to an issue record's *change package*.

#### Opening an Edit Form Tab

To create a new issue record, choose *Issues > New Issue* from the GUI main menu, or click the  button in the GUI main toolbar.

To modify an existing issue record, choose *Issues > Look Up* from the GUI main menu, or click the  button in the GUI main toolbar.



AccuWork prompts you to enter the issue record's unique issue number (an integer).

#### Notes:

##### *Multiple depots*

A typical AccuRev *repository* contains multiple *depots*. Each depot can have its own AccuWork configuration. In each depot, the issue records are numbered sequentially: 1, 2, 3, ... The *New Issue* or *Look Up* command applies to the *current depot*.

##### *Alternate lookup field*

AccuWork can be configured to use a lookup field other than the default (field name *issueNum*, field label *Issue*). In this case, the label for the configured field appears in the prompt window, and you must enter a value that exactly matches (except for uppercase/lowercase) some issue record's value of that field.

## Edit Form Tab Layout

All AccuWork edit forms have these features in common:

- The edit form consists of one or more pages, which appear as subtabs within the overall GUI tab. Typically, there's a header section, which always remains visible as you switch from page to page, using the mouse.

The screenshot shows an AccuWork edit form with a header section at the top containing fields for Short Description, Issue, Status, State, Severity, and Priority. Below the header is a row of subtabs: Basics, Assignment, Misc, Attachments, Resolution, Changes, and Issue History. The Assignment subtab is currently selected, showing fields for Assigned To, Date Assigned, Target Release, and Due Date. Red arrows point to the subtabs and the header section, with labels "multiple 'pages' appear as subtabs" and "header section" respectively.

- Each field is represented by a simple text box, a multiple-line text box, a multiple-choice listbox, or a more complex user-interface "widget". To the left of this widget is a descriptive label for the field. (A red label indicates a required field.)

The screenshot shows a single page of an AccuWork edit form with fields for Submitted By, Date Submitted, Interested Customer, and Category. The Date Submitted field is a complex widget with separate boxes for year, month, day, hour, minute, and second. The Category field is a dropdown menu.

## Special Subtabs

Every edit form has two special subtabs, which appear and work differently than the standard fill-in-the-blanks "pages":

- The *Changes* subtab displays the issue record's *change package*, and provides commands for analyzing and manipulating the change package's entries.
- The *Issue History* subtab displays a complete breakdown of how the issue record's field values have changed over time.

## Working in an Edit Form Tab

You can enter and change values in any number of fields, and visit different pages in a multiple-page form. Here are some notes, describing aspects of AccuWork edit forms that may differ from other fill-in-the-blanks forms:

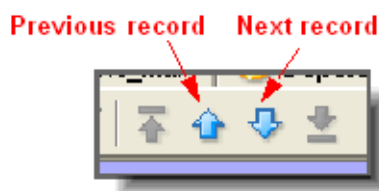
1. Navigating fields and pages

To move from field to field, use the mouse or press **Tab** and **Shift-Tab**. To move from page to page in a multiple-page edit form, click the subtab name (there is no keyboard accelerator).

The **Enter** key confirms a menu choice or enters a line-break in a multiple-line text field. It does not perform field navigation. Within a multiple-line text field, the **Tab** key inserts a TAB character, rather than jumping to the next field.

2. Browsing multiple issue records

While you're working with issue record, browse arrows are enabled in the edit form's toolbar.



This makes it easy to view a set of consecutive issue records. If you've made some changes that you haven't yet saved, AccuWork prompts you to save or discard those changes before switching to the previous or next one.

3. Field initializations

When an edit form appears after you invoke the *New Issue* command, you may notice that some fields already have values. AccuWork can include any number of such field initializations.

4. Required fields

If a field's label appears in **red**, it is a required field. You cannot **Save** the issue record until every required field has a *value*. (A text field has a value if it's non-empty. A multiple-choice listbox field has a value if it's not "<none selected>".) Some of those fields may have gotten their values through an automated field initialization.

If you attempt to save the issue record when one or more required fields still need a value, a pop-up window prompts you to supply the value(s).

5. Other field validations

Field initializations and required fields are implemented through AccuWork's field validations (or "edit checks") facility. As you work in an edit form, changing field values, you may notice these field-validation effects:



- A text field, multiple-choice listbox field, or timestamp field has automatically been set to a specified value.
- The set of required fields has changed.
- A listbox's set of choices has changed.

- The entire issue record, a particular page (subtab), or a particular field has been set to read-only status.

#### 6. An issue record's unique identifier

Each issue record has an integer identifier, its issue number. This number is unique within the depot (but not within the entire repository). AccuWork assigns the issue number automatically the first time the issue record is *Save'd*, and it cannot be changed thereafter. The issue number is stored in field *issueNum*. By default, the label for this field in an edit form is *Issue*, but the edit-form designer can change the label for this field.

### Saving or Discarding Your Work

Your entries in an edit form are not stored in the issue record until you click the  *Save* or  *Save & Close* button in the edit form toolbar. If you don't wish to save your work, just close the edit-form tab as you would any GUI tab.

When you save an issue record:

- (for a new issue record) AccuWork assigns the next available integer issue number to the new issue record. This number cannot be modified.
- The issue record is stored in the depot. A transaction of kind dispatch records the creation or modification of the issue record. No matter how the issue record is modified subsequently, you can revisit the issue record in its current form on the Issue History subtab.
- If you clicked *Save*, the edit form tab remains open, with the newly assigned issue number replacing the "New Issue" label and appearing in the Issue field.

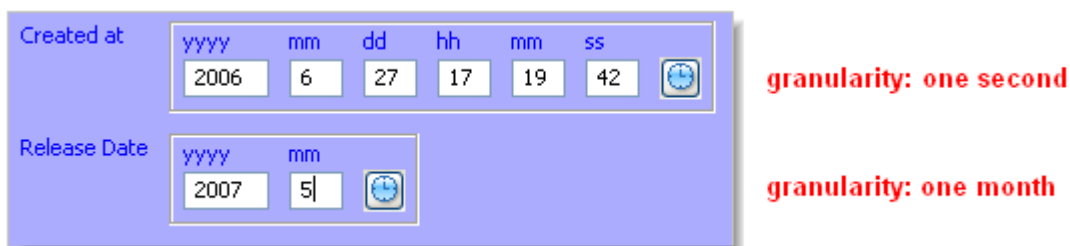
Note: "Issue" is the default label for the issueNum field, but it might have a different label, depending on how the schema is configured.


- If you clicked *Save & Close*, the edit form tab closes.

### Working with Special Fields

AccuWork supports special field types, which have special user-interface "widgets" on an edit form. These are described in the following sections.

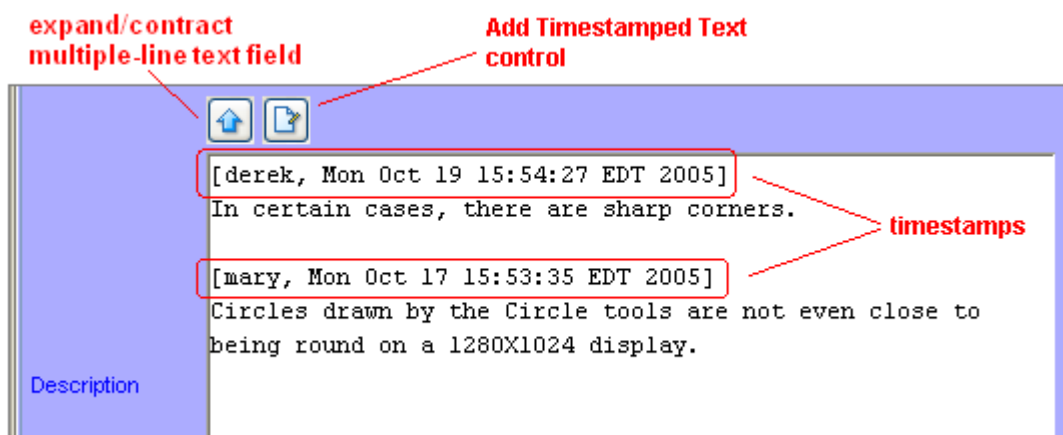
#### Timestamp field






You can fill in the individual subfields manually, or use the  *Select Date* button to display choices ("tomorrow", "2 days ago", etc.) that fill in the subfields automatically. Once these subfields are filled in, you can revise them individually or by clicking the *Select Date* button again.

## Multiple-Line Text and Log fields

A text field can have multiple lines, and can also specify both the width and the height of the textbox on the edit form.



When you use the form you can use the  control at the top of the textbox to temporarily expand the height. Click the control again (it's now ) to contract the height. A scroll bar appears on the textbox whenever the current contents of the field are not completely visible.

A log field is a variant of a multiple-line text field. You can type directly into such a field. And if you click the  *Add Timestamped Text* control, a timestamp text line is inserted at the beginning or end of the field.

## Timespan field

A timespan field is AccuWork's only numeric field type. You can type any value into the field, but when you leave the field, AccuWork automatically erases a non-numeric value. Negative numbers and decimals are allowed. If you enter an integer (e.g. **45**), it's automatically converted to a decimal (**45.0**).

## Attachments field

An edit form can contain one or more attachment fields. In each such field, you can specify one or more files and/or Internet addresses (URLs) to be attached to the current issue record. AccuWork displays the attachments data as a table.

In addition to specifying the location of a file or Internet resource, you enter a name and optional comment. AccuWork automatically fills in your username, the date, and the size of the attached file. (Internet URLs are assigned a size of 0.) If the edit-form field is not large enough to show all these attachment parameters, use the scroll bar to see all the data. You can also resize and rearrange the columns of an attachments table.

An attachment field includes its own toolbar, with these buttons:

### New Attachment

Define a new file attachment for this issue record. A *New File Attachment* dialog appears, in which you can specify one or more files, a name for the attachment, and a comment string.

You can specify multiple files at once -- each one becomes a separate file attachment. In this case, the Attachment Name input field is disabled; each filename is automatically assigned as the attachment name. The comment string that you specify is assigned to each file attachment.

### **New URL**

Define a new attachment to be the address (URL) of an Internet resource. A *New URL Attachment* dialog appears, in which you specify a URL, a name for the attachment, and a comment string. AccuWork inserts the string *http://* in the URL field. You can erase this if you want to specify a location accessed by another Internet protocol, such as *ftp://*.

### **Open Attachment**

Open the existing attachment that is currently selected, using the appropriate program.

### **Save Attachment As**

Create a copy of the currently selected attachment on the client machine.

### **Properties**

Launch a Properties window, displaying the definition of the currently selected attachment. You can use this window to change the attachment's Name or Comment value.

### **Delete Attachment**

Remove the attachment from the issue record (and from the depot). A URL attachment can be deleted at any time. A newly added file attachment can be deleted if you haven't yet *Save'd* the issue record.

When you save the issue record, each file is copied to the depot, so that the data always remains available through the issue record, even if an original file is deleted.

### **Relationship field**

AccuWork supports several kinds of relationships between a pair of issue records:


- **Duplicate:** You can specify that issue record B duplicates issue record A, so that no work need be done on B. (Perhaps the same bug was reported twice.) Any number of records (B, C, D, ...) can duplicate a given record (A). The default AccuWork issue schema requires that a Duplicate relationship be established when an issue record's State field is set to the value **Duplicate**.
- **Dependency:** You can specify that issue record B depends on issue record A (for example, from a project management viewpoint). AccuRev imposes no further semantics on such dependencies; this is a good application for user-defined scripts or practices.

IMPORTANT NOTE: This relationship is independent of *change package dependency* relationships among issue records, which are maintained automatically by AccuRev.

- **Subtask:** You can specify that issue record B defines a subtask of the work to be performed for issue record A. In this case, AccuRev considers issue record A to be "in" a particular stream only if issue record B is "in" that stream, also.

You cannot create "relationship chains": for example, if B duplicates A, you cannot make C a duplicate of B.

Issue record relationships can be viewed only in the AccuWork GUI. The edit-form tables that display issue relationships are not included when you export an issue record (for viewing or printing). These tables are not included when you use the AccuWork command line interface (CLI) to dump the contents of an issue record. (see the [AccuWork™ Command-Line Interface](#) chapter in the AccuRev [CLI User's Guide](#).)

When you create or delete a relationship between issue records, the change to both issue records is saved immediately. There is no need to invoke the *Save* command on the issue record. (The  *Save* button is not enabled in the edit-form toolbar.)

## Viewing and Maintaining an Issue Record's "Duplicate" Relationships

The "duplicate" relationship is not symmetrical: "A duplicates B" is not the same as "B duplicates A".

**in issue record #1**

This issue is duplicated by:

Issue	Short Description
5	Tabs get stuck
6	RGB to HSV conversion is off
12	PDF bookmarks don't work

Total items: 3

This issue duplicates:

Issue	Short Description
-------	-------------------

adding this entry ...

Total items: 0

**in issue record #6**

This issue is duplicated by:

Issue	Short Description
-------	-------------------

... automatically creates this entry

Total items: 0

This issue duplicates:

Issue	Short Description
1	Startup is too slow

Total items: 1

Accordingly, an edit-form field displaying a duplicate-type relationship consists of two tables: the top table shows issue records that are duplicated by the current record; the bottom table shows issue records that the current record duplicates.

Because of the "relationship chains" restriction, issue records can appear in only one of these tables, not both.

On the other hand, "A duplicates B" is the same as "B is duplicated by A". This means that when you add an entry to the *top* table of one issue record, AccuWork automatically adds it to the *bottom* table of the other issue record (and vice-versa).

A Duplicate relationship field has its own toolbar, with these buttons:

## Link Issue

Create a relationship link with another issue record. AccuWork prompts you to enter an issue record number.

## Remove Link

Remove the selected relationship link.


*Note: Duplicate Relationship and Duplicate State*

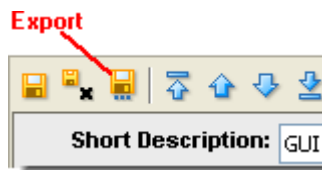
In AccuWork's default issue schema:

- There is a State field in the **header section**, one of whose values is Duplicate. (The Status value must be Closed to enable the setting of the State value to Duplicate.)
- There is a Duplicates field in the Relationships subtab.
- An edit-form **validation** specifies that when you set the State field's value to Duplicate, you must indicate which issue record is duplicated, by creating an entry in the Duplicates field.

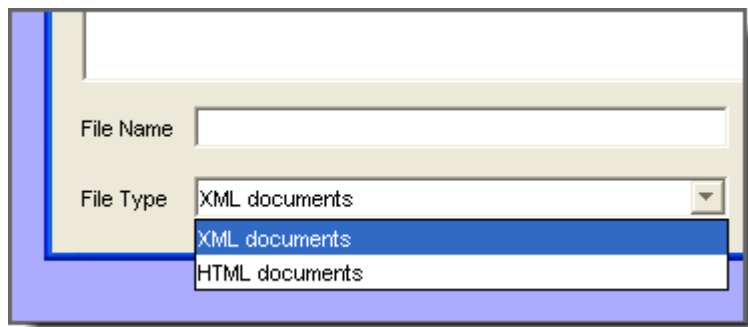
## Exporting an Issue Record (Printing)

At any time when you're using an edit form, you can print the issue record. For AccuWork, printing means "publish to the Web" -- which means either "create an HTML file" or "create an XML file".

1. Click the  Export button in the form's toolbar.



2. Select the file type in the File Chooser dialog that appears.



3. Specify a pathname for the export file. You don't need to specify the .html or .xml suffix -- AccuWork adds it automatically.

When creating an HTML file, AccuWork outputs all the content of the issue record and approximates the form layout, too. Even if the edit form has multiple pages, you need to "print" only once. All of the pages are combined into a single HTML or XML file:]

**AccuRev Dispatch Issue: 12**

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Work Google

Address: http://localhost:1000/

AdSubtract Stats Ads Jira Pops 100%

Basics [Assignment] [Attachments] [Resolution]

Short Description	Issue	Status	Subscribed	State	URL
FLT bookmarks don't work	12	Open			
Severity	3	Priority	3		

[Back to top](#)

**Basics**

Product	Platform	Type	default
Platform	All	Sub-System	<none selected>
Phase Bound In	Production	Bound In Release	roll C
Submitted By	mary	Date Submitted	yyyy mm dd hh mm ss 2003 9 5 17 33 42
Interested Customer	Category	<none selected>	

[Back to top](#)

**Assignment**

Assigned To	mary	Date Assigned	yyyy mm dd hh mm ss 2003 9 25 9 19 1
Target Release	roll C	Date Data	yyyy mm dd

My Home

On Windows machines, AccuWork automatically invokes a web browser on the HTML file it creates. If you wish, resize the browser window to optimize the look of the issue record. HTML documents automatically adjust to changes in window width. Then, use the browser's print command to create a printout of the issue record.

## The Edit Form Tab -- "Changes" Subtab

The Changes subtab of an *issue record*'s edit form shows the *change package* of the issue record. You can invoke commands to see the details of the change package's changes, maintain its entries, and view the transaction and version history of its elements.

For a detailed discussion of change packages, see [Patches and Change Packages](#) on page 172.

### "Changes" Subtab Layout

The Changes subtab contains a table, each row displaying one "change package entry". A given element can have at most one entry in a change package -- but this entry can "grow" over time, as development work continues on the element. The table includes these columns:

### Element (or separate "Name" and "In Folder" columns)

The element's pathname within the depot.

### Version

The *head version* of the change package entry.



### Basis Version

The *basis version* of the change package entry.

The head version and basis version are the "endpoints" of the change package entry. That is, the entry consists of all the versions between these two versions. (The head version is included; the basis version isn't.)

## Working in the "Changes" Subtab

The Changes subtab has commands for modifying change package contents:

- The  *Remove* command deletes one or more selected entries from the change package.
- The  *Send to Issue* command copies one or more selected entries to another issue record's change package.


But the change package facility is designed so that most modifications to change packages occur as "side effects" to the *Promote* command, by the change-package-level integration between AccuRev and AccuWork. ([Change-Package-Level Integration between AccuRev and AccuWork](#) on page 328.)

See also: [Working with Tables](#) on page 9

## Commands Available in the "Changes" Subtab

### Remove

Deletes the selected entry(s) from the change package.

This change takes effect immediately -- you cannot discard the change by closing the edit form without invoking the *Save* command. (Removing change package entries does not enable the  *Save* button in the edit-form toolbar.)

### Diff Against Basis

(text-file elements only) Compares the entry's head version with the basis version. This shows exactly what changes to this element are included in the change package.

### Show History

Open a History Browser tab, containing the transactions involving the selected element.

### Browse Versions

Open a Version Browser tab, showing all the versions of the selected element, and their interrelationships (ancestry).

### **Send to Issue**

Copies the selected entry(s) to another issue record's change package. If there's an existing entry for an element in a destination change package, an attempt is made to combine the existing entry with the entry you specified. The change(s) take place immediately in the destination issue record(s). There is no way to cancel the change(s) -- but you can open the issue record(s) and *Remove* the entry(s).

The *default query* is executed, and the results are displayed in a dialog (see [The Send to Issue Command](#) on page 129). You are prompted to choose one or more of the issue records selected by the query. You can also create a new issue record, whose number will be entered in the dialog.

### **Properties**

Displays information about the selected element. The data items displayed vary with the type of element.

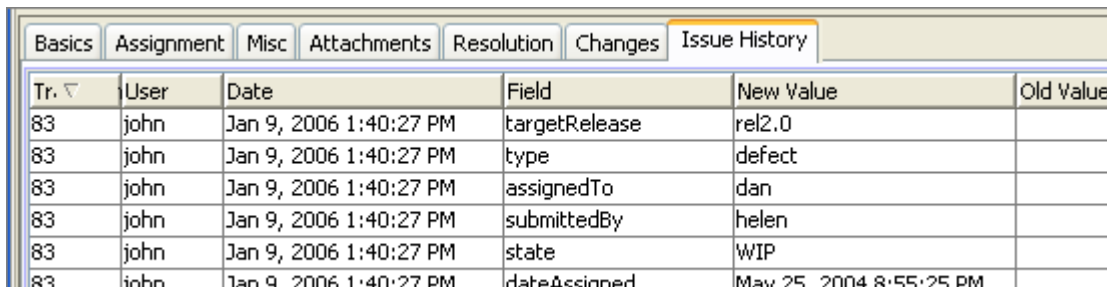
## The Edit Form Tab -- "Issue History" Subtab

The edit form of an AccuWork issue record always includes a subtab named *Issue History*, which is maintained automatically by AccuWork. This subtab shows, on a field-by-field basis, how the issue record has changed over time.

You can use this subtab to view past "versions" of the issue record.

### "Issue History" Subtab Layout

The Issue History subtab contains a table, each row of which shows how one field was changed (or initially created) by a particular *Save* of the issue record.



Tr. ▾	User	Date	Field	New Value	Old Value
83	john	Jan 9, 2006 1:40:27 PM	targetRelease	rel2.0	
83	john	Jan 9, 2006 1:40:27 PM	type	defect	
83	john	Jan 9, 2006 1:40:27 PM	assignedTo	dan	
83	john	Jan 9, 2006 1:40:27 PM	submittedBy	helen	
83	john	Jan 9, 2006 1:40:27 PM	state	WIP	
83	john	Jan 9, 2006 1:40:27 PM	dateAssigned	May 25, 2004 8:55:25 PM	

The table contains these columns:

#### **TransNum**

The number of the *Save transaction*. (The transaction kind is **dispatch**.)

#### **User**

The *user* who made the change to the issue record.

#### **Date**

The time at which the change took place.

**Field**

The field whose value changed (or was initially created).

**New Value**

The value to which this transaction changed this field.

**Old Value**

The value of this field prior to the *Save*. (Empty for the issue record's initial *Save* transaction.)


**Working in an "Issue History" Subtab**

If you double-click any row of the Issue History table, AccuWork opens a read-only edit form, showing the state of the issue record at the time of that particular *Save*. (Exception: the issue record's change package does not participate in this feature.)

## The New File Attachment Dialog

The Browse for Attachment dialog allows you to browse for files to attach to an *issue record* when in a field of type Attachments.

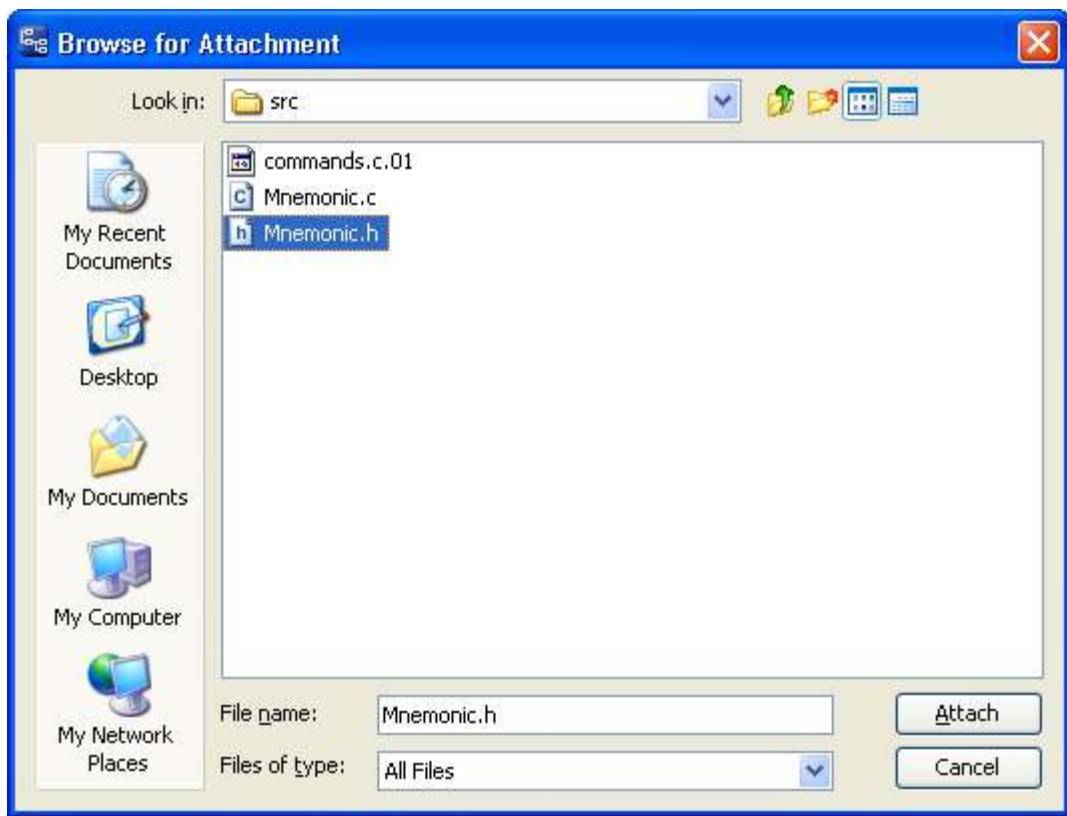
**Invoking the New File Attachment Command**

Click the  New File Attachment icon in the toolbar of the Attachments field in an issue record's *edit form*.

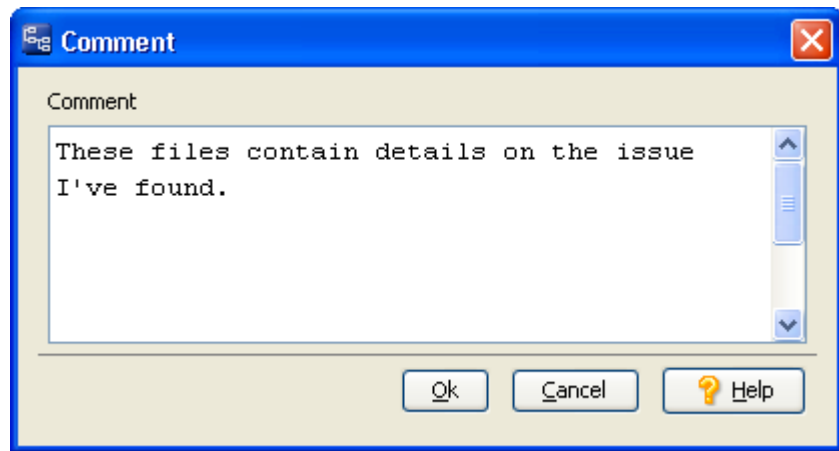
**Using the Browse for Attachment Dialog**

The dialog is a standard file-open dialog for your operating system.

Browse to a directory, choose one or more files, and click *Attach*.




Then enter comments (if desired), in the Comments dialog that displays. Click *OK* to attach the file(s) to the issue record.



## The New URL Attachment Dialog

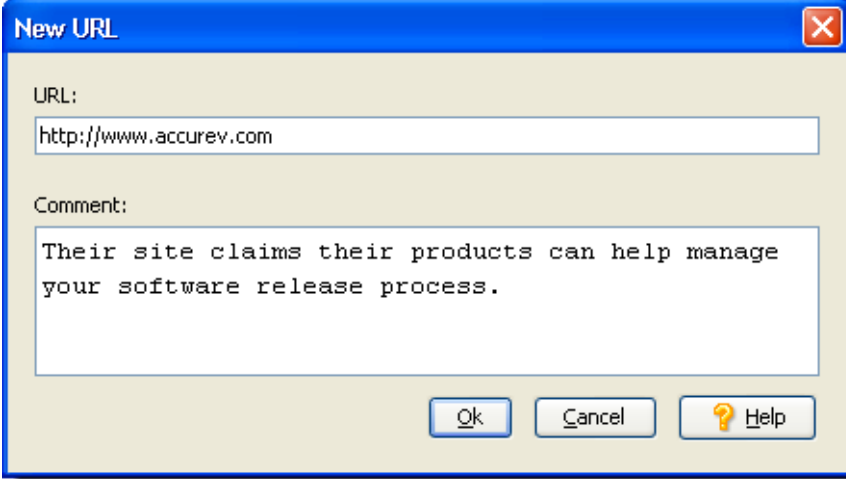
The New URL Attachment dialog defines one or more URL attachments in an *issue record* field of type Attachments.

### Invoking the New URL Attachment Command

Click the  New URL Attachment icon in the toolbar of the Attachments field in an issue record's *edit form*.

## Using the New URL Dialog

Fill in these fields to define one or more attachments:

A screenshot of a 'New URL' dialog box. It has a blue title bar with the text 'New URL' and a red close button. The dialog contains two input fields: 'URL:' with the text 'http://www.accurev.com' and 'Comment:' with the text 'Their site claims their products can help manage your software release process.' At the bottom right are three buttons: 'Ok', 'Cancel', and 'Help' (with a question mark icon).

### URL

A full Web address.

### Comment

Any text string.


## Issue Record Queries

### The AccuWork Queries Tab

An AccuWork query retrieves a set of records according to user-defined selection criteria. AccuWork has a point-and-click interface for creating and editing queries; it enables you to create simple queries quickly, and to create sophisticated queries in a straightforward, reliable way.

By default, the queries that you create are private queries, which cannot be seen by other users. You can declare any query to be a public query. Such queries are visible to all users, who can use and copy them, but cannot modify your original.

### Opening a Queries Tab

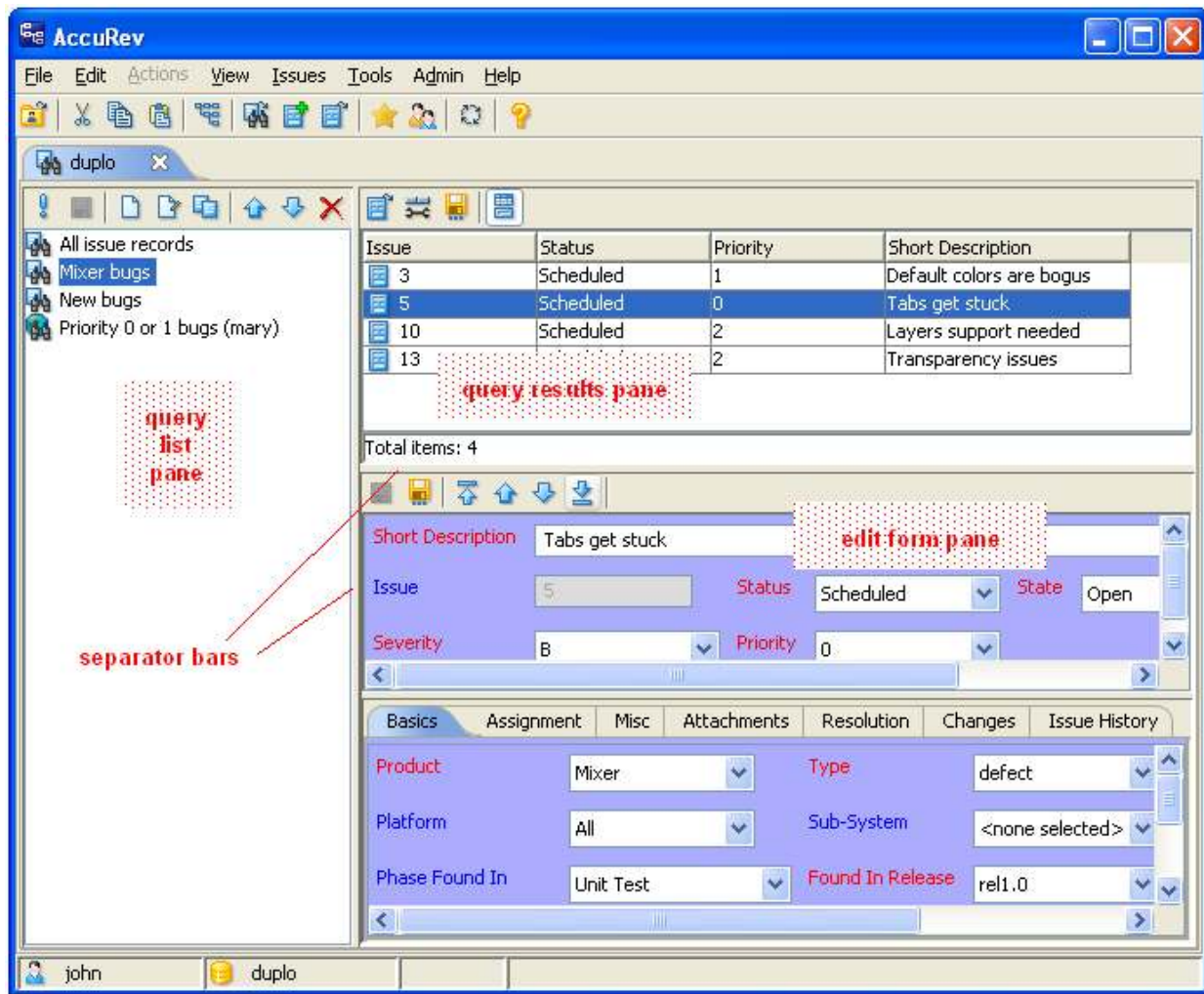
Choose the *Issues > Queries* command from the GUI main menu, or click the  button in the GUI main toolbar.




A *Queries* tab opens.

## Query Mode Layout

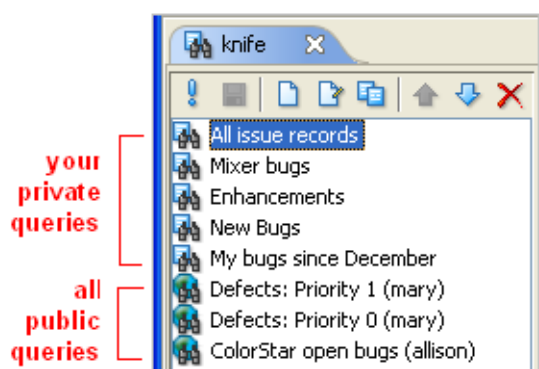
In Standard Query mode, the Queries tab includes two (and sometimes three) panes, each with its own toolbar:



- The *Query List* pane lists the names of all your existing *private queries*, along with *public queries* available to all users. Using the toolbar in this pane, you compose new queries, view/revise/rearrange/execute existing queries, and delete queries.
- When you execute a query, the set of issue records selected by the query are displayed as a results table in the *Query Results* pane. If you've set a *default query* for this depot, it's executed automatically when you open the Queries tab.
- If you click the  *Show Issue Form* button in the toolbar of the Query Results pane, an *Edit Form* pane appears below it. This pane is a fully functional edit form, which you can use to view and modify the issue record currently selected in the Query Results pane. A user preference (*Tools > Preferences*) configures this pane to appear automatically when you open a Queries tab.

## Query List Pane Layout

The Query List pane contains a list of the *private queries* you've defined for the current depot, along with any *public queries* defined by you and/or other users.



The listing is not alphabetical -- whenever you create a new query, it's simply added to the end of the list.

Your queries (both private and public) are always listed above other users' public queries.

See [Working in the Query List Pane](#) on page 291.

## Query Results Pane Layout

The Query Results pane contains a table that displays the results of a query.

Issue	Status	Priority	Short Description
10	Scheduled	2	Layers support needed
11	New	3	Polylines need more controls
14	Scheduled	3	Need to support new output types

Each row of the table displays one issue record; each column displays a particular issue-record field.


Each query has its own results-table design: a set of columns (fields), in a particular order and with particular column-widths. The results table's design can also include a single-level or multiple-level sort order for the rows (*issue records*).

When you create a new query, AccuWork automatically starts off the results table with a couple of columns, including Issue (**issueNum** field). You can add more columns at any time. You can also remove the Issue column from the table. **Note:** "Issue" is the default label for the issueNum field, but the edit-form designer can change the label.

See [Working in the Query Results Pane](#) on page 293.

## Edit Form Pane Layout

The Edit Form pane is just like the "full size" *Edit Form* tab, except that it occupies a pane within the Queries tab, instead of having an entire tab to itself. This pane displays the entire contents of the issue record that's currently selected in the Query Results pane. In addition, browse arrows are

enabled in the Edit Form pane's toolbar, making it easy to view some or all of the records selected by a query. There is also a *Change Package History* icon (  ) that displays all the modifications that have been made to the change package for this issue. In the upper-right corner, the **Issue URL** link allows you to easily capture a URL to the current issue, using the WebUI. To configure this feature, create or edit:

```
<accurev_install_dir>\storage\site_slice\dispatch\config\settings.xml
```

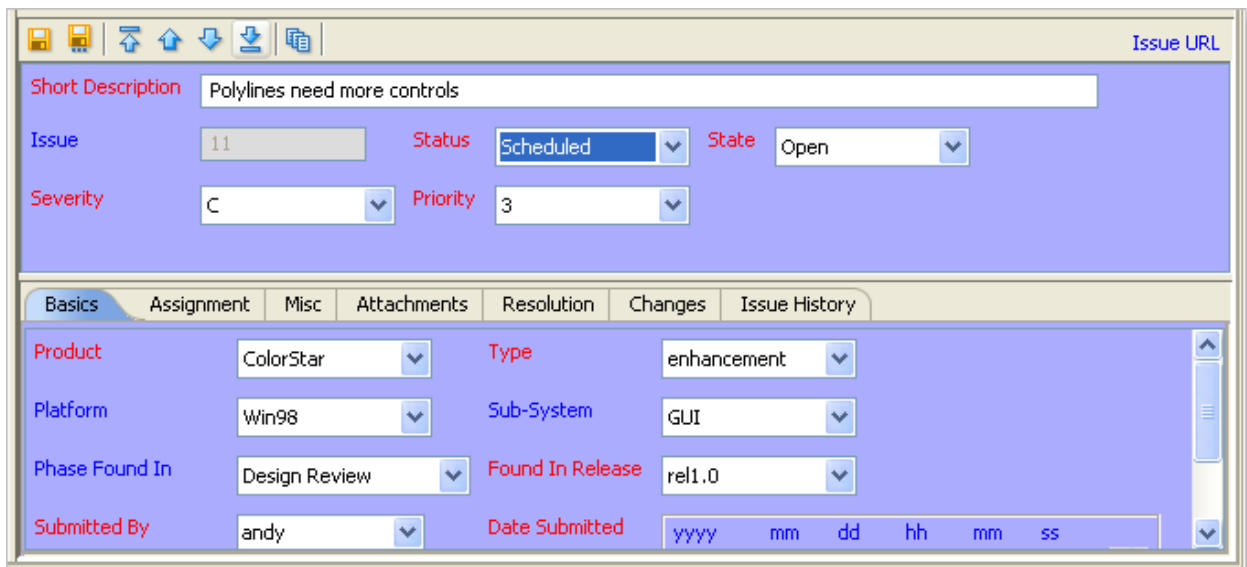
to contain the following entry (customized for your specific installation) to point to your WebUI server:

```
<settings>
```

```
  <webui url="http://<webui-host>:<port>/accurev"/>
```

```
</settings>
```

To use this feature, either click **Issue URL** to bring up the issue in the WebUI, or to simply copy the URL to send to somebody else, right-click **Issue URL** and then select and copy the URL from the resulting dialogbox.



See [Working in the Edit Form Pane](#) on page 294.

## Working in Query Mode

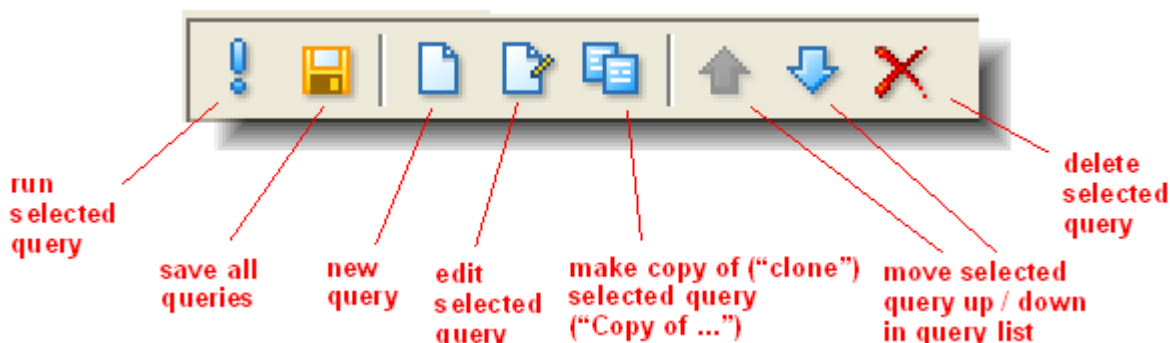
Here's a typical Queries tab procedure:

1. In the Query List pane, create a new query (or select an existing one), then execute the query.
2. In the Query Results pane, view selected fields from the issue records selected by the query.
3. Open the Edit Form pane to view the entire contents of one of the selected issue records (and perhaps modify it).

The following sections describe working in each of the Queries tab's panes.

## Working in the Query List Pane

You can invoke the following commands in the Query List pane, using context menus or the Query List toolbar. Most commands operate on the currently selected query.



### Run query

(equivalent: double-click the query name) Executes (or re-executes) the selected query, displaying the results in the Query Results pane. You don't need to run this command when you revise a query in the *Edit Query* window -- AccuWork automatically executes the revised query and updates the results table.

### Save all queries

Save all your private queries. (There is no way to save a single query.) Your private queries are stored in the AccuRev repository within the depot directory, in an XML-format file:

```
.../storage/<depot-name>/dispatch/config/user/<username>/  
query.xml
```

#### **Notes:**

Each user's private queries are stored separately

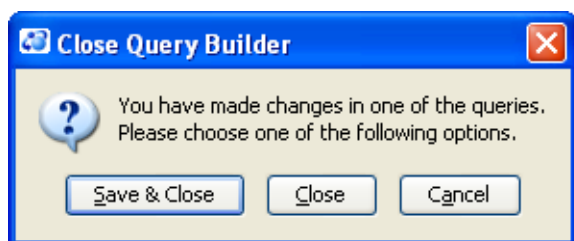
The <username> directory in the query storage pathname causes the private queries for each user to be stored separately.

Queries not version-controlled

Although your queries are stored in the depot, they are not version-controlled in the way AccuRev files are. For example, there is no command that displays or reinstates your queries as you saved them two days ago.

Automatic save-on-close feature

If you try to close a Queries tab in which you've made changes to one or more queries, but haven't run the Save all queries command, AccuWork offers to perform the save.



### **New query**

### **Edit query**

Create a new query or revise an existing one, using the *Edit Query* window.

### **Clone query**

Create a private query that is a copy of the selected private or public query. The new query is initially named "Copy of ...", but you can change the name.

### **Move query up**

### **Move query down**

Change the position of the selected query in the Query List pane. Note that your own queries (both private and public) are always listed above other users' public queries.

### **Delete query**

Remove the currently selected query. The deletion does not take effect until you run the *Save all queries* command. If you close the Queries tab without saving your queries, the "deleted" query will still exist the next time you open the Queries tab. This may or may not be the right thing to do: closing the Queries tab without saving also discards changes you've made to other queries since the most recent save.

### **Set as Default**

#### **Disable as Default**

Designates one of your private queries to be -- or to stop being -- your *default query* for this depot. (This is your default query -- another user can set one of his own private queries to be his default query.)

The query name is redisplayed in ***bold-italic*** to indicate that it is the default query.

Your default query is executed:

- Whenever you open a new Queries tab.
- When the *transaction-level integration* between AccuRev and AccuWork is triggered by the *Promote* command (see [Transaction-Level Integration between AccuRev and AccuWork](#) on page 331).
- When you execute [The Send to Issue Command](#) on page 129.

A query loses its status as your default query when you run *Disable as Default* on it, or when you select another query as the default.

### **Set as Private**

### **Set as Public**

Changes a query that you created from public to private, or vice-versa.

### **Additional Operations in the Query List Pane**

- Renaming a Query—You can rename any of your private queries in the Query List pane, without having to open an Edit Query window:
  1. Select the query.



2. Press function key F2 or click the query name again. (Be careful not to double-click; that invokes the Run Query command.)
- Revisiting the Results of a Query—Whenever you select a particular query in the Query List pane, the Query Results pane displays that query's most recent results. The cache of previous query results is cleared when you close the Queries tab.

## Working in the Query Results Pane

In many cases, browsing the results table produced by running a query may be all you need to do. But in other cases, you may want to see a selected issue record in the context of its edit form. Why?

- The results table is read-only. To modify an issue record, you must access it through its edit form.
- The data you need may be in a field that is not included in the results table. Instead of adding a column to the results table, you can use the edit form to display all the fields.

You can use a "full-size" or "reduced-size" edit form to view a issue record:

- Invoke the  *Open Issue* command to open a separate *Edit Form* tab on the currently selected issue record.
- Invoke the  *Show Issue Form* command to make an *Edit Form* pane appear within the Queries tab.

With either tool, browse arrows in the edit form's toolbar make it easy to view some or all of the records in the results table.

## Commands Available in the Query Results Pane

The following commands are available in the Query Results pane:



### Open Issue

Open a separate *Edit Form* tab to view/revise the currently selected issue record.

### Setup Columns

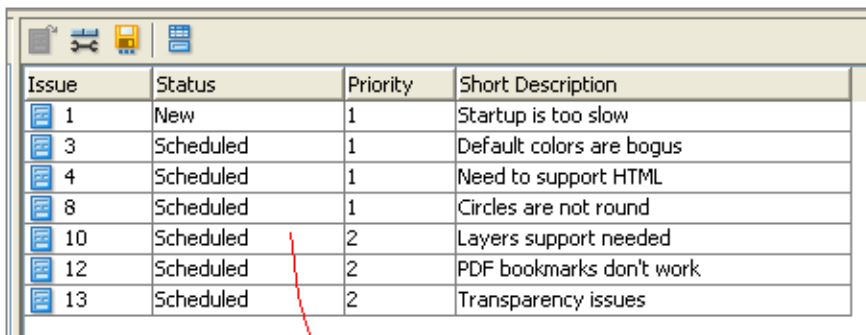
Open a *Columns window*, in which you can change which columns (fields) appear in the table, and also change their order.

### Remove Column

(appears only in the context menu of an individual column) Removes the column you right-clicked from the results table. (This is a shortcut, eliminating the need to invoke the *Setup Columns* command.)

## Export Table

Open a *File Chooser* window, in which you specify a file to store the entire contents of the results table:



Issue	Status	Priority	Short Description
1	New	1	Startup is too slow
3	Scheduled	1	Default colors are bogus
4	Scheduled	1	Need to support HTML
8	Scheduled	1	Circles are not round
10	Scheduled	2	Layers support needed
12	Scheduled	2	PDF bookmarks don't work
13	Scheduled	2	Transparency issues

 "print" to HTML file

### Priority 1 or 2

Issue	Status	Priority	Short Description
1	New	1	Startup is too slow
3	Scheduled	1	Default colors are bogus
4	Scheduled	1	Need to support HTML
8	Scheduled	1	Circles are not round
10	Scheduled	2	Layers support needed
12	Scheduled	2	PDF bookmarks don't work
13	Scheduled	2	Transparency issues

Total items: 7

HTML file displayed in Web browser

## Show Issue Form

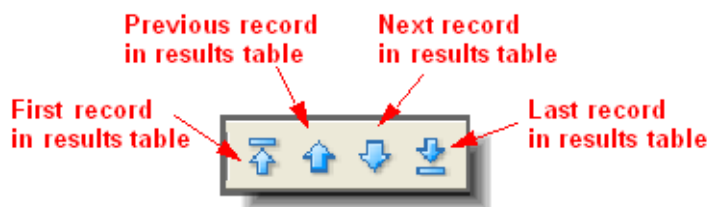
Toggles whether an *Edit Form* pane appears, displaying the complete contents of the currently selected record in the Query Results pane.

See also: [Working with Tables](#) on page 9.

## Working in the Edit Form Pane

Using a "reduced-size" edit form within a pane of the Queries tab is virtually the same as using a "full-size" edit form in its own tab. Here are some notes on the few differences:


- The browse buttons in the Query Edit Form pane's toolbar makes it easy to visit all the records in the Query Results pane table.

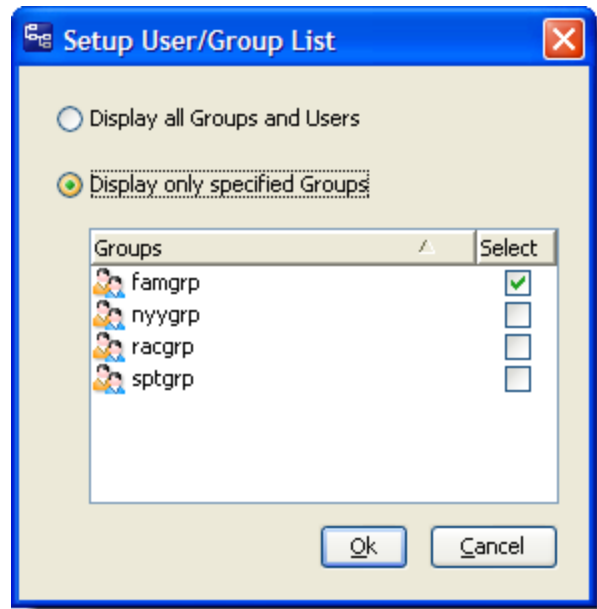


- After modifying a record and invoking the  Save command, you may want to rerun the query in order to update the table in the Query Results pane. This doesn't occur automatically.


## Configuring the Users/Groups and Streams Panes

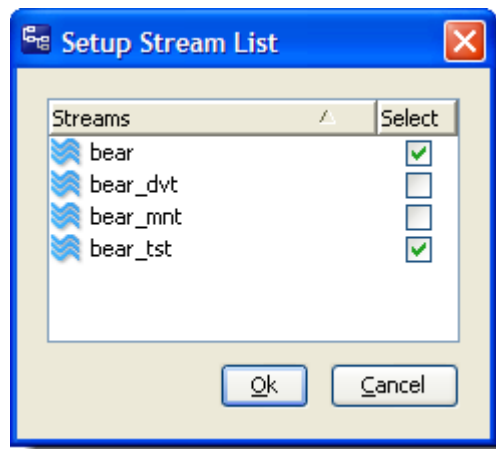
You can choose which user and group names appear in the users/groups pane, and which stream names appear in the streams pane:

- Use the  **Setup User/Group List** toolbar button to bring up a dialog box in which you select the items to appear in the users/groups pane.



In this example, only the group name **famgrp** and the name of each member of **famgrp** will appear in the users/groups pane.

- Use the  **Setup Stream List** toolbar button to bring up a dialog box in which you select the items to appear in the streams pane.







- In this example, only the stream names **bear** and **bear\_tst** will appear in the streams pane.

# The AccuWork Query Editor

The AccuWork Query Editor is a tool for creating and revising AccuWork queries.

## Opening the Query Editor

You can create or revise an AccuWork query in these contexts:

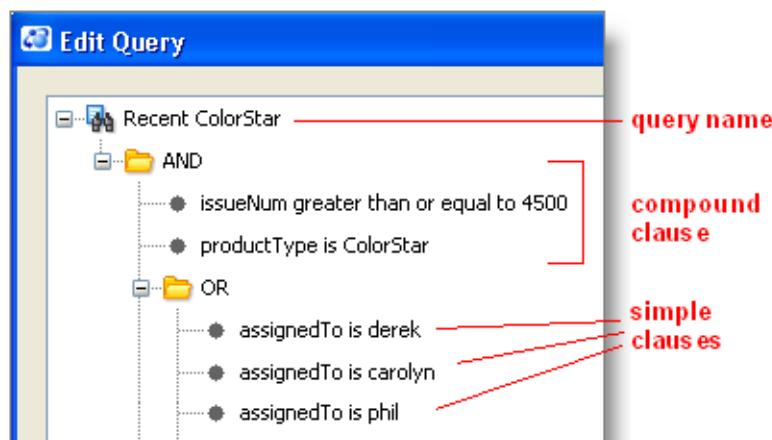
- In a *Queries* tab, click the  *New Query* or  *Edit Query* button. This opens a separate Edit Query window.
- In the Change Packages subtab of a Schema Editor tab, click the  *New Trigger* or  *Edit Trigger* button to open a separate Promote Trigger window. The Query Editor runs in the bottom pane of this window.

## Query Editor Layout

The Query Editor displays the contents of one query. The query appears as a hierarchy, which you navigate using familiar expand/collapse controls. The hierarchical organization is a natural fit, because each query is, itself, a hierarchy of simple clauses and compound clauses:

- The first hierarchy level contains the query's name, if the Query Editor was invoked from the Queries Tab.
- A simple clause, such as value of the assignedTo field is mary, can live at any level.
- A "compound AND" or "compound OR" clause consists of:
  - The operator AND or OR at one hierarchy level
  - Any number of clauses (simple or compound) at the next hierarchy level.

For example, this query can be expressed as:




"Retrieve each issue record that is (1) numbered at or above 4500, (2) applies to product ColorStar, and (3) is assigned to user *derek*, *carolyn*, or *phil*."

The context menu of a query's name in the Query Editor includes the command *In English*, which produces a sentence like the one above.

## Working in the Query Editor

The following sections describe the details of working in the Query Editor.

### Naming a New Query / Renaming an Existing Query

When you click the  *New Query* button to create an empty query, AccuWork assigns it a placeholder name ("New Query *nnn*"). You can edit the name now: click the name once to select it, then press function key **F2** (or click it a second time) to begin editing it. Don't double-click — that's equivalent to using the expand/collapse control.

You can change a query's name at any time while Query Editor is still active. Or you can change it later, in the *Query List* pane.

### Creating a Simple Clause

Every query consists of at least one *simple clause*. A simple clause has three parts:

*<field-name> <comparison-operator> <value>*

The point-and-click interface makes creating a simple clause easy and (almost) foolproof. Start by clicking one of the "Click here to add ..." placeholders in the query:

First, you must select the *<field-name>* part of the clause from the list-box containing all the field-names. When you select a field-name, the query editor automatically adjusts the *<comparison-operator>* and *<value>* parts of the clause, based on the selected field. In the example below, the user has selected field-name *productType*, whose value must be one of these names: *ColorStar*, *Mixer*, *Publisher*.

The table below shows all the AccuWork data types, along with the corresponding choices for the *<comparison-operator>* and *<value>* parts of a simple clause. As you "fill in the blanks" to create simple clauses, you'll notice that AccuWork allocates new "Click here to add ..." placeholders, so that one is always available at each level of the query.

Field Type	Comparison Operator	Value
------------	---------------------	-------

Text	contains matches does not contain does not match equal to not equal to less than less than or equal to greater than greater than or equal to	<p>Any character string. (Do not enclose it in quotes.) The value is always interpreted as a string literal; there is no way to specify the value of some other field here.</p> <p>The comparison is always a case-insensitive string comparison, never a numeric comparison. For example, the value <b>3</b> is greater than the value <b>25</b>.</p> <p>The <i>contains</i> and <i>does not contain</i> operators perform a simple substring search (case-insensitive), with no pattern matching.</p> <p>The <i>matches</i> and <i>does not match</i> operators perform pattern matching (case-insensitive), using these wildcards:</p> <ul style="list-style-type: none"> <li><b>*</b> matches 0 or more characters</li> <li><b>?</b> matches any 1 character</li> <li><b>[ækz]</b> matches <b>a</b>, <b>e</b>, <b>k</b>, or <b>z</b></li> <li><b>[a-e]</b> matches <b>a</b>, <b>b</b>, <b>c</b>, <b>d</b>, or <b>e</b> (<b>Note:</b> Don't mix uppercase and lowercase; neither <b>[E-k]</b> nor <b>[e-K]</b> matches any character)</li> <li><b>{one,two,seven}</b> matches <b>one</b>, <b>two</b>, or <b>seven</b></li> <li><b>\</b> or <b>/</b> Matches any directory-separator character, even if it's the "wrong one" for the client machine. For example, <b>src/do_*.java</b> matches <b>src\do_something.java</b> on a Windows client.</li> </ul> <p><b>Examples:</b></p> <p><b>Rls 4*</b> matches <b>Rls 4</b> and <b>Rls 4.01</b>, but not <b>Rls 4.2</b> or <b>Rls 5</b></p> <p><b>Rls [2-5]*</b> matches <b>Rls 2</b>, <b>Rls3.04</b>, and <b>Rls 5.3</b>, but not <b>Rls 1.02</b></p> <p><b>the -? option</b> matches <b>the -X option</b> and <b>the -4 option</b>, but not <b>the -Op option</b></p> <p><b>cop{y,ies}</b> matches <b>copy</b> and <b>copies</b>, but not <b>copy(s)</b></p>
Timespan	equal to not equal to less than less than or equal to greater than greater than or equal to	<p>A numeric value, representing an amount of time.</p> <p>Note: users specify the value in the edit form as a number of <i>hours</i> (e.g. 7.5); an XML-format dump of the issue record created by the <i>Export</i> command reports the value as a number of <i>minutes</i> (e.g. 450).</p>
Choose	is is not	One of the strings specified in the definition of this field in the Schema Editor.
List	is is not	One of the strings specified in the definition of a particular named list in the Schema Editor.
User	is is not is member of is not member of	One of the principal-names in the user registry maintained by the AccuRev server. Alternatively, a user-group defined in the registry.

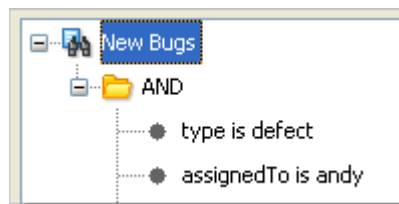
Timestamp	is is not is before is after is before or equal to is after or equal to	An AccuRev timestamp.
Attachments	contains	Any character string. This string is compared to the <i>Name</i> of each of an issue record's attachments. See .
internal	equal to not equal to less than less than or equal to greater than greater than or equal to	An integer, identifying a particular AccuWork issue record ( <i>issueNum</i> field) or a particular AccuRev transaction ( <i>transNum</i> field).

### Creating a Compound Clause

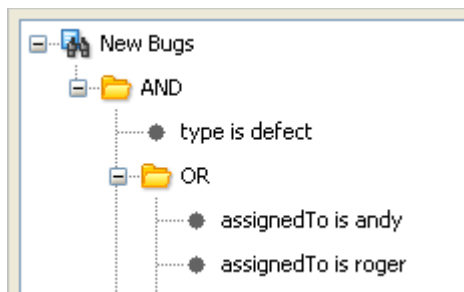
A *compound clause* combines any number of subclauses together, using the same logical operator: AND or OR. (The NOT operator is not supported.) The subclauses to be combined can, themselves, be either simple or compound.

#### **Examples and notes:**

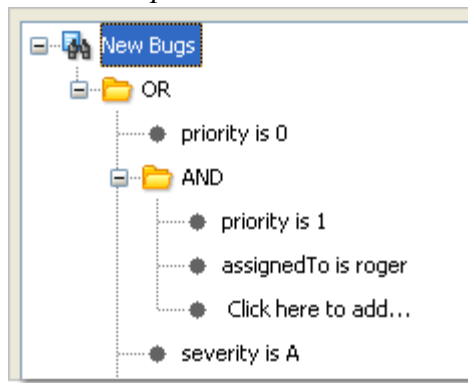
*Example 1: simple AND simple*



*Example 2: simple AND compound*

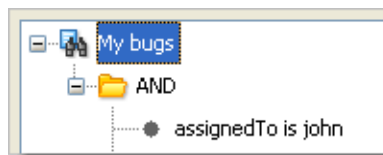


*Example 3: simple OR compound OR simple*



*Example 4: Compound clause with a single subclause*

A compound clause can contain a single subclause. This is logically equivalent to using the subclause by itself. In fact, the standard "one-line" query contain an AND clause with one simple subclause.

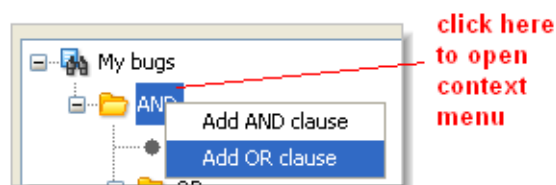


The query editor automatically creates placeholders for simple clauses. But you must explicitly insert a compound-clause placeholder yourself, then fill in the subclauses.



Following are guidelines for adding placeholders to a query:

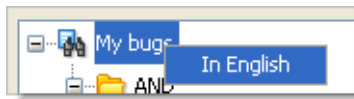
- You cannot add a compound clause at the top level (directly beneath the query name). This level contains a single compound clause -- initially AND, but changeable to OR .
- To create a compound clause below a particular operator, right-click it, and select the type of clause from the context menu.



- To change any AND clause to an OR clause (or vice-versa), select the operator and press function key F2 (or click it a second time). This opens a listbox with the values AND and OR.

## Viewing an English-Language Rendition of the Query

The query name's context menu includes the command *In English*, which renders the query in everyday language, displaying the results in a separate window.



## Ending the Query Editor Session

Whether you're composing a new query or revising an existing one, you end by saving your work (*Ok* button) or discarding it (*Cancel* button). If you didn't cancel the Query Editor session, AccuWork automatically executes the query and displays the results in the Query Results pane.

**Note:** Closing the window using its title bar controls is equivalent to *Cancel*.

## The Setup Columns Command

The *Setup Columns* command specifies a set of fields from AccuWork, and the order in which they will appear as columns in a particular table.

**Note:** *What about column widths?*

You don't specify column widths with this command. After finishing this command, you can drag column separators in the tab from which you invoked *Select Columns*. You can also drag-and-drop the column headers to rearrange the columns.

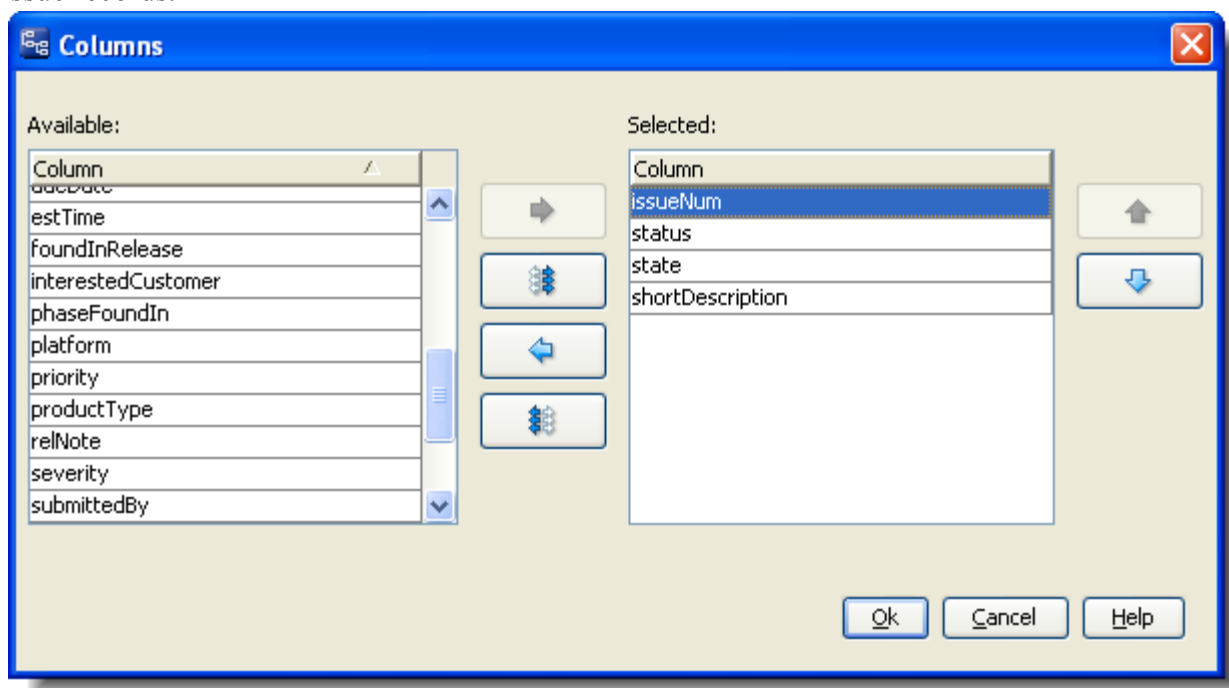
## Invoking the Setup Columns Command

A *Setup Columns* command button appears in these AccuWork contexts:

- Schema editor: Change Packages tab (Change Package Results pane)  
Specifies the format of the tables produced by Show Issues and Show Diff by Issues commands.
- Schema editor: Change Packages tab (Change Package Triggers pane)  
Specifies the format of the table displayed by the change-package-level integration between AccuRev and AccuWork.
- Schema editor: Relationship Types tab  
Specifies the format of the table that shows the set of issue records related to a given issue record.
- Queries tab: Query Results pane toolbar  
Specifies the format of the table that shows the results of a user-defined AccuWork query.

## Using the Setup Columns Dialog

The fields listed under **Selected** when you click *Ok* will be displayed in the appropriate table of issue records.



- To move field names from the Available list to the Selected list (or vice-versa), select one or more names, then click the ➡ button (or the ⬅ button).
- To move all field names between the lists, click the ↔ or ↔ button.
- To rearrange the Selected list, select one or more names, then click the ↑ or ↓ button.

### Notes:

#### *Field names vs field labels*


The names that you work with in the Columns window are the actual **field names** (Name column of the Schema subtab of the AccuWork Schema Editor) In the Query Results pane, the column headers are the corresponding **field labels** (Label column of the Schema subtab).

#### *Further adjustments, including column-width specifications*

You don't specify column widths in the Columns window, just their order. You can adjust column widths and change the order of columns: see [Working with Tables](#) on page 9.

#### *Multiple-line text fields*

You can include a multiple-line text field in the set of columns to be displayed in the Query Results pane, but it's likely that you won't be able to see the complete contents of such fields, no matter how wide you make its column. But when you display an HTML file created by the *Export Table* command in a web browser, you'll see the complete contents of every field.

When you click *Ok* to close the Setup Columns dialog, the changes you've made are reflected immediately in the Query Results pane. But the changes to the query's columns setup are not saved in the repository until you invoke the  *Save All* command in the Query List pane.

## Schema Editor

### The AccuWork Schema Editor -- Overview

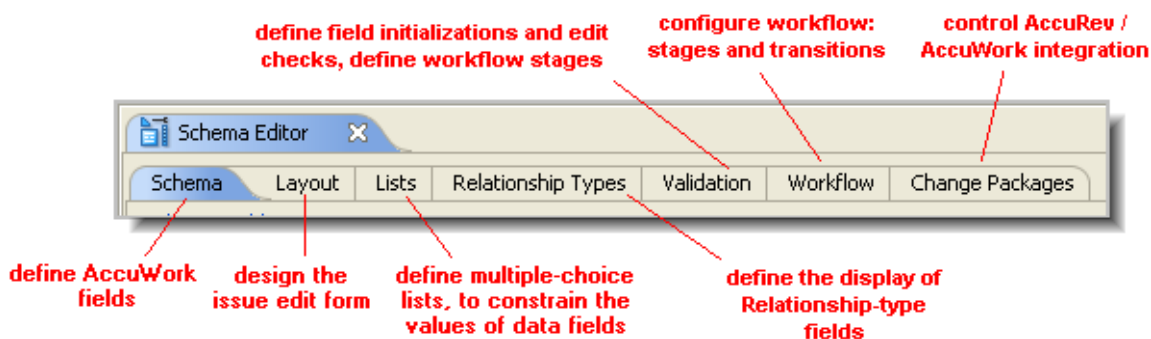
Using the AccuWork Schema Editor, you can configure AccuWork, including *issue record* layout and validation. Each *depot* can contain a separate AccuWork configuration, including a custom-designed *edit form*, through which users create and modify the issue records. You can make the edit form "smart" by defining *validations* (edit checks) that specify default values, required fields, and interrelationships among multiple fields.

#### Invoking the Schema Editor

Choose *Admin > Schema Editor* from the GUI main menu. The first time you invoke this command in a particular depot, AccuWork offers to use the repository's default schema. Accepting this offer copies a set of XML-format configuration files from the *site\_slice/dispatch/config* subdirectory to this depot.

**Note:** The default schema does not actually become the schema for this depot until you click the Schema Editor's Save button.

The Schema Editor tab includes these subtabs:



#### Saving Changes to the Schema

At any time while working in the Schema Editor, you can click the *Save* button in the lower right corner of the Schema Editor tab. This saves the current state of the schema to a set of XML-format files in subdirectory *dispatch/config* of the depot directory (slice) in the AccuRev repository :

- Contents of the Schema subtab: *schema.xml*
- Contents of the Layout subtab: *layout.xml*
- Contents of the Lists subtab: *lists.xml*
- Contents of the Relationship Types subtab: *relation\_types.xml*

- Contents of the Validation subtab: logic.xml
- Contents of the Change Package Results section of the Change Packages subtab: cpk\_fields.xml
- Contents of the Change Package Triggers section of the Change Packages subtab: cpk\_promote\_queries.xml

## The AccuWork Schema Editor

### (Schema subtab)

The Schema tab of the Schema Editor contains a table that details all the data fields defined for AccuWork in the current depot.

The screenshot shows the 'Schema Fields' subtab of the Schema Editor. At the top, there are tabs for 'Schema', 'Layout', 'Lists', 'Relationship Types', 'Validation', and 'Change Packages'. Below the tabs, there is a '3pty ITS Key:' dropdown menu set to 'shortDescription'. The main area contains a table with the following data:

Name	Type	Label	Report Width
actTime	Timespan	Actual Time	10
affectedFiles	Text	Affected Files	10
assignedTo	User	Assigned To	15
Attachments	Attachments	Attachments	10
category	Choose	Category	10
closedBy	User	Closed By	15
closedInRelease	List	Closed in Release	15
dateAssigned	Timestamp	Date Assigned	18
dateClosed	Timestamp	Date Closed	15
dateCompleted	Timestamp	Date Completed	15
dateSubmitted	Timestamp	Date Submitted	15
description	Text	Description	40
designInfo	Text	Design Information	40
dueDate	Timestamp	Due Date	15
estTime	Timespan	Est Time	10
foundInRelease	List	Found In Release	15
interestedCustomer	Text	Interested Customer	24
issueNum	internal	Issue	10
phaseFoundIn	Choose	Phase Found In	15
platform	Choose	Platform	18
priority	Choose	Priority	10

At the bottom of the window, there is a checkbox labeled 'Show including hidden', and three buttons: 'Add...', 'Remove', and 'Reactivate'.

If you are not using the repository's default schema, AccuWork initializes the Schema Fields table with two fields.

- **issueNum**: An integer-valued field that records the position of the *issue record* in the depot. This number is assigned when a user creates the record (i.e. at the first *Save* on the edit form), and it never changes.

- **transNum:** An integer-valued field that records the *transaction* number of the most recent update to the issue record. The transaction appears in the depot's transaction log -- the same log that records AccuRev transactions, such as *keep* and *promote*. The type of the issue-record-update transaction is **dispatch**.

## Notes:

### *issueNum vs. transNum*

A record's *issueNum* value never changes, but its *transNum* value changes each time a user *Save's* the record. *issueNum* and *transNum* are indexes into two different databases.

### *AccuRev/AccuWork integrations*

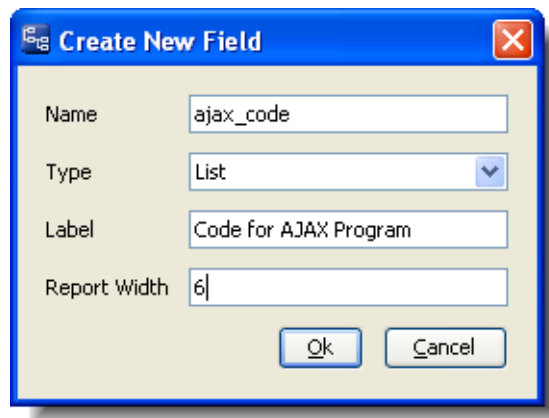
The change-package-level and transaction-level between AccuRev and AccuWork also update issue records, and so changes the value of the *transNum* field. (See [Change-Package-Level Integration between AccuRev and AccuWork](#) on page 328 and [Transaction-Level Integration between AccuRev and AccuWork](#) on page 331.)

See Also: [Schema Editor](#) on page 303

## Adding and Removing Fields from the Schema

You can define any number of additional fields in the schema. Follow these steps for each new field:

1. Click the Add button at the bottom of the Schema subtab.
2. Fill in the Create New Field window that appears:



- Name: The official field-name of the field. AccuWork users don't ever need to know this name - they know the field by its label. The field-name must not contain any SPACE characters. Validations must be expressed in terms of the field's name, not its label.
  - Type: One of the **data types** supported by AccuWork.
  - Label: The field-label character string that identifies the field on the edit form. A field-label can contain SPACE characters (e.g. Last Name ).
  - Report Width: An integer that determines the relative width of the field in the HTML table created when the user clicks Export on the edit form of an individual issue record.
3. Click Ok to close the Create New Field window.

4. In the Field Values box to the right of the Schema Fields table, specify additional information about the field. The kind of information required varies with the **data type**.

The screenshot shows the Schema Editor window with the 'Schema' tab selected. The 'Schema Fields' section on the left contains a table with the following data:

Name	Type	Label
actTime	Text	Actual Time
assignedTo	User	Assignee
completionDate	Timestamp	Completion Date
estTime	Timespan	Estimated Time
finalDisp	Choose	Final Disposition
issueLog	Log	Issue Log
issueNum	internal	Issue
product	Choose	Product Name
submittedBy	User	Submitted By

The 'Field Values' box on the right contains a list with 'Cancelled' and 'Verified' (selected). Below the list are 'Add' and 'Remove' buttons.

Repeat the steps above as often as required to create new fields. Your field definitions are not saved until you click the *Save* button in the lower-right corner of the Schema Editor tab. You cannot save your work until you place at least one field in the edit form (Layout subtab).

### Removing a Field / Restoring a Field

To remove an existing field (except for *issueNum* and *transNum*), select it and click the *Remove* button. The field disappears from the Schema tab and can no longer be used in the edit form. But any data stored in existing issue records is preserved.

When you remove a field from the schema, AccuWork checks whether the field is used in the edit form. If so, it removes the field from the edit form.

You can restore a removed field to the schema:

The screenshot shows the Schema Editor window with the 'Schema' tab selected. The 'Schema Fields' section on the left contains a table with the following data:


Name	Type	Label
description	Text	Description
designInfo	Text	Design Information
dueDate	Timestamp	Due Date
dvt_group	Text	Devel Group
estTime	Timespan	Est Time
foundInRelease	List	Found In Release
interestedCustomer	Text	Interested Customer
issueNum	internal	Issue
phaseFoundIn	Choose	Phase Found In

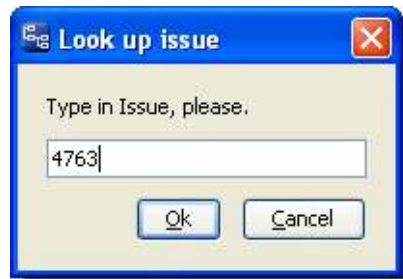
The 'dvt\_group' field is highlighted with a blue background. A red arrow points from the 'Show including hidden' checkbox (labeled 1) to the 'dvt\_group' field. At the bottom right, the 'Reactivate' button is circled in red and labeled 2.

1. Check the *Show Including Hidden* checkbox. All removed fields appear in the list, with a gray background.

2. Select the field to be restored, and click the *Reactivate* button.

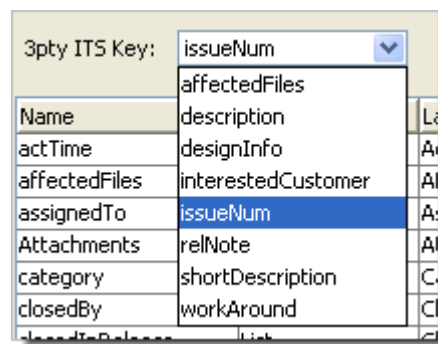
## Specifying the Lookup Field

The  *Issues* > *Look Up* command prompts the user to enter a value, then uses that value to locate an issue record.



By default, this lookup operation uses the **issueNum** field (whose user-visible label is **Issue**). You can configure another field to be used for issue-record lookups, using the *3pty ITS Key* listbox. (This feature is designed to facilitate AccuBridge integrations with third-party issue-tracking systems.)

Select another field from the listbox, and save the schema.



Thereafter, the *Issues* > *Look Up* command will prompt for a value to be matched against the specified field.

### Notes:

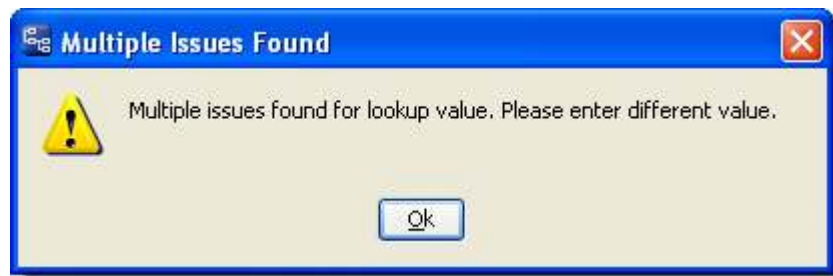
#### *Value-matching on lookups*

The *Issues* > *Look Up* command requires an exact match between the user's entry and the value in the issue-record field. Exception: upper-case/lowercase differences are ignored. This can cause problems if the lookup field is a Text field.

#### *Uniqueness of lookup values*

AccuRev guarantees uniqueness of the *issueNum* field value only. It allows duplicate values in all other fields. Be sure to specify an alternative lookup field whose values are

rendered unique by some other mechanism. If a user performs a lookup on a non-unique value, an error occurs:



## Integrating Configuration Management and Issue Management: the 'affectedFiles' Field and Change Packages

If you wish to enable the integration of a depot's version-controlled files and its AccuWork issue records, define a field whose name is *affectedFiles*. The field's *data type* must be **Text**. You can choose any label for the field. (Such a definition is included in the default AccuWork schema.)

The integration also depends on the enabling of a built-in AccuRev trigger procedure:

```
accurev mktrig -p <depot-name> pre-promote-trig
client_dispatch_promote
```

The integration routine writes the transaction number of each *promote* command to the *affectedFiles* field of a particular issue record. Alternatively, in the AccuRev Enterprise version of AccuWork, the integration routine records each promoted version in the issue record, in a special section named *Changes*. This section is maintained automatically by AccuWork -- you don't need to define any fields to enable this additional aspect of the integration.

See Also:

- [Change-Package-Level Integration between AccuRev and AccuWork](#) on page 328
- [Transaction-Level Integration between AccuRev and AccuWork](#) on page 331

## Data Types

Each field you define in the *Create New Field* window must have one of the AccuWork data types listed in the table below.

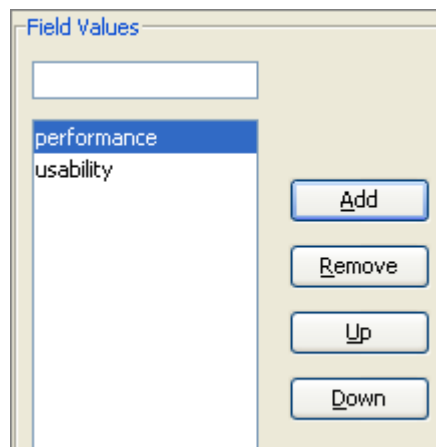
Field Type	Possible Values	Required Additional Information in "Field Values" Box
Text	Any character string. For multiple-line fields, the string can include line-terminators.	<i>Height</i> : number of lines displayed for the field in the edit form (default: 1). For multiple-line fields (height > 1), the edit form includes an expand/contract button to increase/decrease the number of lines displayed. <i>Width</i> : relative width of field in the edit form.
Choose	One of the strings specified in the Field Values box for this field.	A set of strings. In the edit form, a list-box containing this set of strings is offered to the user.

Timestamp	An AccuRev timestamp.	Granularity (year, month, day, hour, minute, or second). In the edit form, the user fills in fields that indicate a time, to the granularity you specify here.
User	One of the principal-names in the user registry maintained by the AccuRev server.	None. In the edit form, a list-box is offered to the user, containing the names of all registered AccuRev <i>users</i> .
Stream	A Project stream name in the current depot.	None. <b>NOTE: THIS DATATYPE WILL BE DEPRECATED AFTER RELEASE 5.4.0</b>
Timespan	A numeric value, indicating a number of hours. Decimal values (e.g. 4.5) are allowed.	None.
List	One of the strings specified in the definition of a particular named list.	The name of an existing list, defined on the <i>Lists</i> subtab. In the edit form, a list-box containing the set of strings defined in the named list is offered to the user.
Log	Any character string (variant of text field type)	In the edit form, an Add Text control appears above the input field. The user can type directly into the input field, or can click the Add Text control and create a "log entry" in the popup window that appears.
Attachments	A set of attachment definitions.	<i>Height, Width</i> : the height (number of lines) and width (approximate number of characters) of the edit-form field that lists the attachment definitions.
Relationship	A set of issue records	See <a href="#">The AccuWork Schema Editor (Relationship Types subtab)</a> on page 317.
Internal	A positive integer.	None. You cannot create a field with this data type; it is used only by the built-in fields <i>issueNum</i> and <i>transNum</i> .

### Defining Multiple-Choice Lists

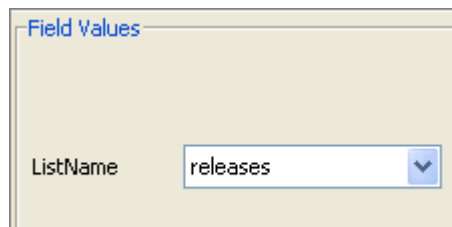
The *choose* and *list* data types are similar:

- For a *choose* field, the set of possible values -- an ordered list of strings -- is part of the individual field definition. You enter the possible-values list in the Field Values box for that field.



The 'Field Values' dialog box for a *choose* field. It features a text input field at the top. Below it is a list box containing the items 'performance' and 'usability', with 'performance' selected. To the right of the list box are four buttons: 'Add', 'Remove', 'Up', and 'Down'.

- For a *list* field, the set of possible values is also an ordered list, but it is *not* part of the individual field definition. In the Field Values box, you specify the name of one of the lists created on the *Lists* subtab. Any number of *list* fields in AccuWork (within a single depot) can use the same named list.



The 'Field Values' dialog box for a *list* field. It shows a label 'ListName' followed by a dropdown menu. The dropdown menu currently displays the value 'releases'.

The mechanics of defining the ordered list is similar in the "local" case (for an individual *choose* field) and in the "global" case (on the Lists subtab, for use by any number of *list* fields). On the Lists subtab, you must supply a ListName for the list; for an individual *choose* field, the possible-values list doesn't need or have a name.

## Defining Issue Record Relationships

When you create a field of type *Relationship*, you must select the *Duplicate*, *Subtask*, or *Dependency* relationship type for the field. On an edit form, the field's edit-widget is a pair of tables. The upper table displays "inward" or "child" relationship links -- that is, links *from* other issue records to the current record. The lower table displays "outward" or "parent" relationship links -- that is, links *to* other issue records from the current record. (Multiple-link chains are not allowed -- each issue record can be related to others by child links or parent links, but not both.)

The figure below show how a *Duplicate* relationship field appears in an edit form. A relationship field of type *Subtask* or *Dependency* appears similarly.

This issue is duplicated by:

Issue	Short Description
-------	-------------------

table of "child" relationship links

Total items: 0

This issue duplicates:

Issue	Short Description
-------	-------------------

table of "parent" relationship links

Total items: 0

choose columns for these tables on the Schema Editor's Relationship Types subtab

## The AccuWork Schema Editor

### (Layout subtab)

An edit form consists of field-labels and corresponding edit-widgets, arranged in rows and columns. The field-labels come from the Labels column of the Schema Fields table (Schema

subtab). The edit-widget for a field depends on its data type and, in some cases, on the additional Field Value information.

The screenshot shows a form layout with two columns and five rows. Red annotations identify components:

- column**: Labels for the two columns at the top.
- row**: Labels for each of the five rows on the left.
- field-label**: Points to the 'Assigned to' label in the first row.
- edit-widget**: Points to the dropdown menu for 'Assigned to' in the first row.
- "Description" field spans two columns**: Points to the large text area in the fifth row.

The form contains the following fields:

- Row 1: Assigned to (dropdown), Date Assigned (date/time picker).
- Row 2: Target Release (dropdown), Due Date (date picker).
- Row 3: Date Completed (date/time picker).
- Row 4: Est Time (text input), Actual Time (text input).
- Row 5: Description (large text area).

## Contents of the Layout Subtab

The Layout subtab provides a drag-and-drop canvas on which you design the edit form's rows and columns. On this canvas, each field-label/edit-widget pair is represented by a yellow box.

The screenshot shows the 'Layout' subtab of a software interface. It features a drag-and-drop canvas with yellow boxes representing field-label/edit-widget pairs. Red annotations provide details:

- field-names used, not field-labels**: Points to the 'actTime' box in the 'Development' section.
- edit-widget that spans multiple columns**: Points to the 'issueLog' box in the 'Testing' section.
- fields defined on Schema subtab, but not placed on edit-form**: Points to 'completionDate' and 'finalDisp' in the 'Fields' list.
- colors to be used on edit-form**: Points to the color selection area on the right, which includes 'Foreground' (blue), 'Background' (light blue), and 'Required Foreground' (red).

The interface includes tabs for Schema, Layout, Lists, Relationship Types, Validation, and Change Packages. The 'Layout' tab is active, showing a canvas with sections like 'Issue Fields', 'Development', and 'Testing'.

As the screen-shot annotations indicate, the yellow box contains a field-name (e.g. **actTime**), not a field-label (**Actual Time**). On the edit form, the field-label and the corresponding edit-widget will appear, side by side, at this position. You can also expand a yellow box to make it span two or more columns.

You can organize the fields into multiple "tables", each of which defines a separate tabbed page in the edit form. One of the tables can define a header section, which always appears on the edit form, no matter which tabbed page is currently visible.

The screenshot shows a web-based issue edit form. At the top, there's a header section with fields for 'Issue:' (value: 4), 'Stage:' (value: New), 'Summary:' (value: Need to support HTML), 'Submitted By:' (value: allison), and 'Product Name:' (value: Publisher). Below this is a tabbed interface with four tabs: 'Development' (selected), 'Testing', 'Changes', and 'Issue History'. At the bottom, there are fields for 'Assignee:' (value: allison) and 'Target Release:' (value: brick). Red annotations point to the header section and the tabbed pages.

The layout for a multiple-page edit form looks like this:

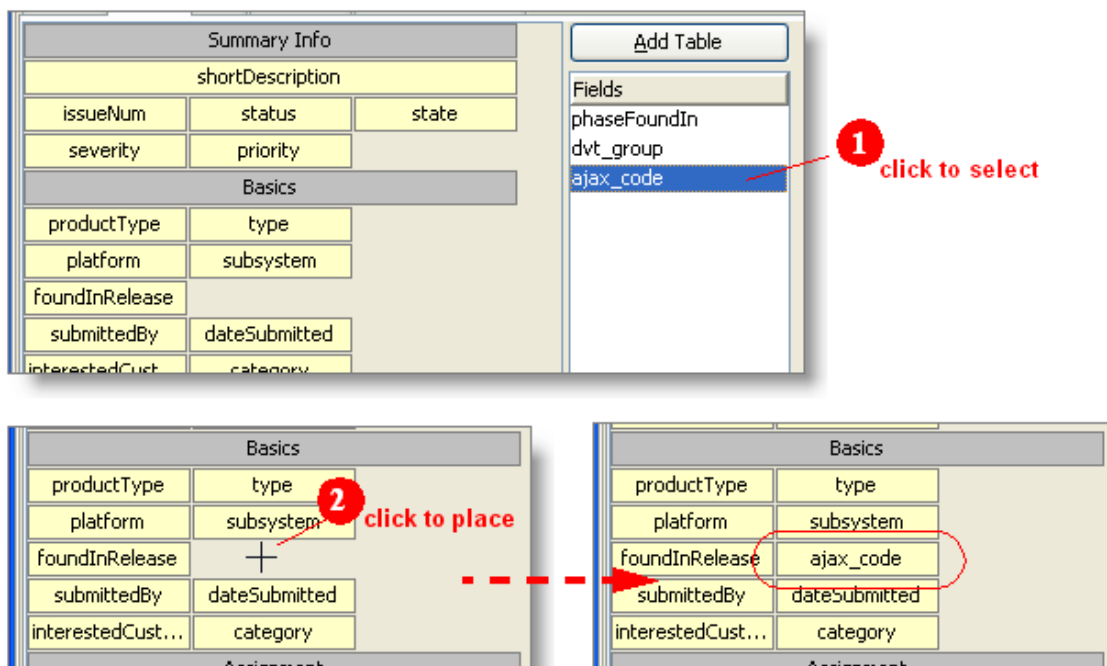
The screenshot shows the 'Layout' subtab of the form design tool. It displays a hierarchical structure of fields organized into tables. The 'Issue Fields' table contains 'issueNum' and 'wflowStage'. The 'summary' table contains 'submittedBy' and 'product'. The 'Development' table contains 'assignedTo', 'targetRelease', 'estTime', and 'actTime'. The 'issueLog' table contains 'issueLog'. The 'Testing' table contains 'testedBy' and 'completionDate'. Red annotations point to each table, stating that each table defines a separate tabbed page on the edit form.

## Form Layout Operations

This section lists the edit-form design operations available on the Layout subtab.

## Adding a field to the edit form

The *Fields* list-box offers all fields that do not currently appear in the edit form. Click to select a field in this list-box. Then, click at the empty location on the canvas where you want to place the field.

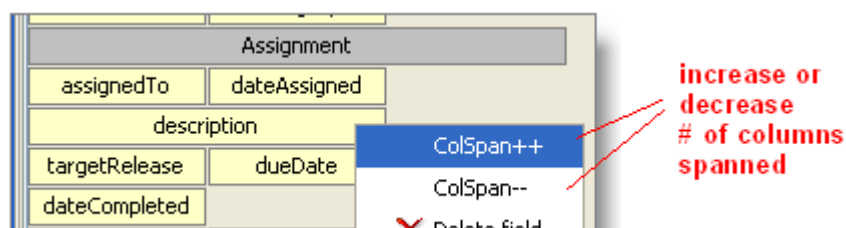


## Arranging fields into rows and columns

Drag-and-drop the yellow boxes to the desired location on the canvas. Drop a box on top of another one to push it rightward or downward.

## Changing column-spanning

Right-click a yellow box, and select one of the spanning operations from the context menu.



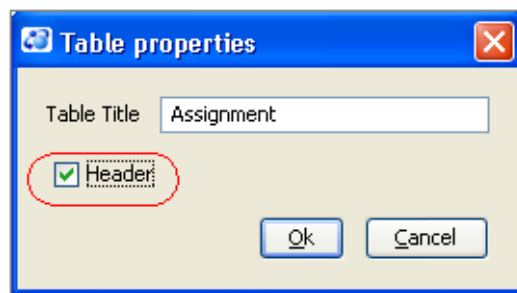
## Creating a new page in the edit form

Click the *Add Table* button above the *Fields* list-box. Fill in the Table Title field in the dialog box that appears, then click *OK* . This title will appear on the page's tab in the edit form.

A gray box with the specified title appears at the bottom of the layout. Fields that you place below this box will appear on the new page of the edit form.

### Creating a header table for the edit form

When creating a new page for the edit form (see above), check the *Header* checkbox if you want it to be the header table.



An edit form can have at most one header table. If another table is currently selected as the header table, it is automatically deselected. Its fields always appear at the top of the edit form, even if it is not the topmost table on the Layout tab.

You can also redefine an existing page to be a header table: right-click the gray box and select *Properties*. All fields in the header table will appear at the top of the database's edit form, above every tabbed page.

### Renaming a page

Right-click on the gray box, select *Properties*, and change the entry in the Table Title box.

### Arranging fields into multiple pages

Drag-and-drop the yellow boxes (fields) and the gray boxes (pages). Fields that you place below a gray box will appear on the corresponding page of the edit form.

### Removing a field or page

Right-click the yellow box (field) or gray box (page), and select *Delete* from the context menu. Deleting a page doesn't delete the fields currently on the page -- it just deletes the gray box, effectively merging the fields into the page above. **Note:** You cannot delete the first gray box; an edit form must have at least one page.

### Setting the edit form's colors

Click one of the colored buttons at the right side of the Layout subtab. Then select a color from the *Pick a color* window. You can set these colors:

- Foreground: the color of the edit form's field-labels.
- Required Foreground: the field-label color for fields where a value must be specified. Required fields are defined on the Validations subtab.
- Background: the color of the edit form's background.

# The AccuWork Schema Editor

## (Lists subtab)

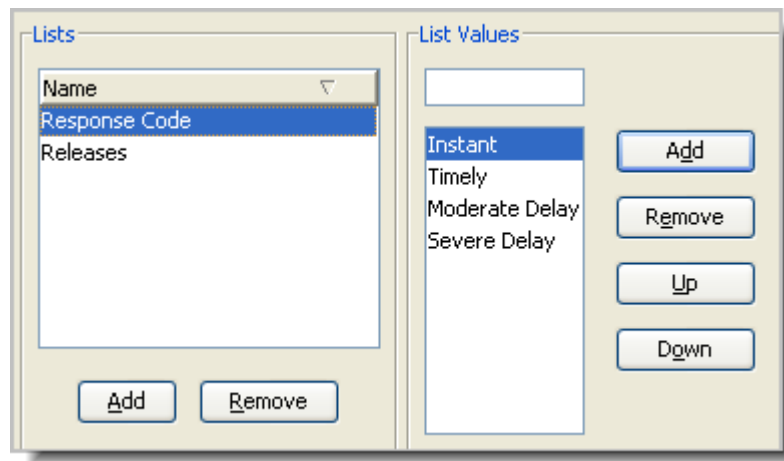
A field whose data type is **List** can have any of the values in a predefined list. Any number of fields can get their values from the same list.

The Lists subtab maintains the set of predefined lists for AccuWork within a given depot .

### Lists Subtab Layout

The Lists subtab has two sections:

- In the "Lists" section, you create and delete named lists.
- In the "List Values" section, you specify the values in each list.



### Lists Subtab Operations

this section describes the edit-form design operations available on the Layout subtab.

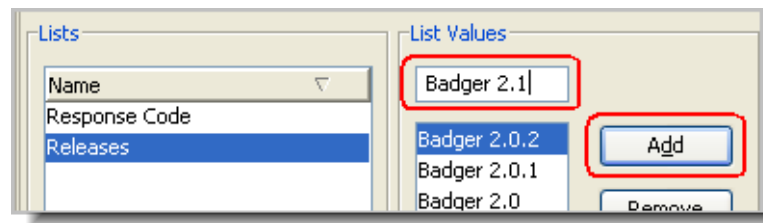
#### Creating a New List

Click the *Add* button in the Lists section, and type a name into the popup window.



## Specifying the Values for a List

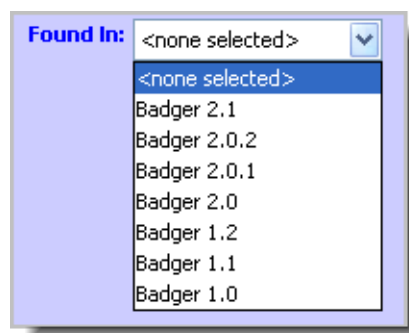
Select one of the list names in the Lists section. Then, in the List Values section: type a name in the input field at the top, and click the *Add* button.



The new name is added to the listbox below the input field, allowing you to type another name and click *Add* again.

## Reordering the Values in a List

Select a name in the listbox showing the current values, and click the *Up* or *Down* button.



The order of the names in this listbox determines the order in which they will appear in the edit-widget on an edit form.

## Removing a List

In the Lists section: select a name, then click the *Remove* button.

## Renaming a List

There is current no provision for renaming an existing list.

# The AccuWork Schema Editor

## (Relationship Types subtab)

An issue record can be related to one or more other issue records, through a variety of relationship types:

- **Duplicate:** You can specify that issue record B duplicates issue record A, so that no work need be done on B. (Perhaps the same bug was reported twice.) Any number of records (B, C, D, ...) can duplicate a given record (A). The default AccuWork schema requires that a

Duplicate relationship be established when an issue record's State field is set to the value **Duplicate**.

- **Dependency:** You can specify that issue record B depends on issue record A (for example, from a project management viewpoint). AccuRev imposes no further semantics on such dependencies; this is a good application for user-defined scripts or practices.

IMPORTANT NOTE: This relationship is independent of *change package dependency* relationships among issue records, which are maintained automatically by AccuRev.

- **Subtask:** You can specify that issue record B defines a subtask of the work to be performed for issue record A. In this case, AccuRev considers issue record A to be "in" a particular stream only if issue record B is "in" that stream, also.
- **Track:** This relationship type is used internally by AccuWork to track change package gaps.

The edit-widget for a field whose *data type* is **Relationship** is a table. It displays selected fields from all the other issue records that are related, through a particular relationship type, to the current record.

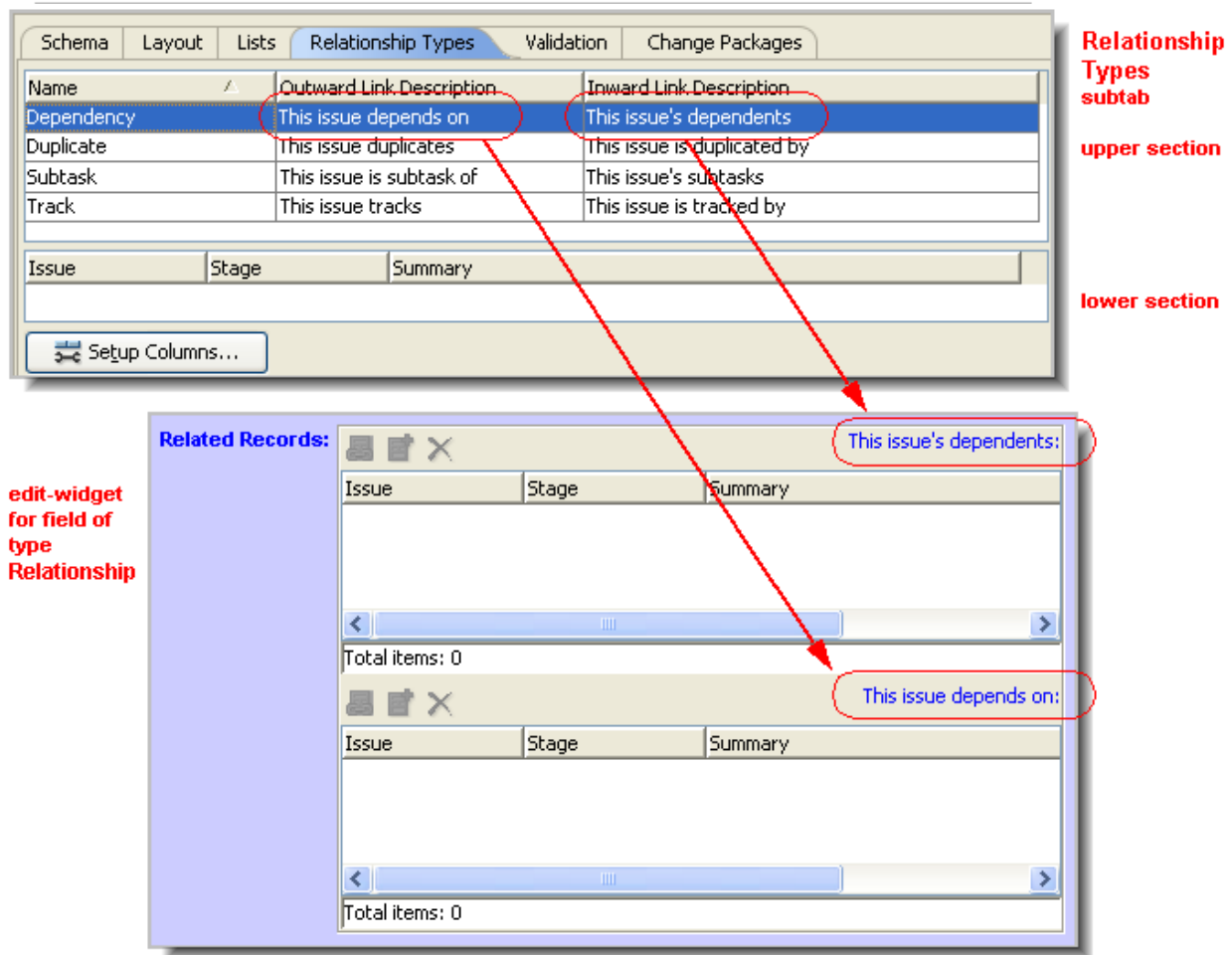
On this subtab, you specify -- for each relationship type -- which fields from a set of related issue records are to be displayed in a **Relationship** field.

## Relationship Types Subtab Layout

The Relationship Types subtab contains two sections:

- The upper section lists the several relationship types, along with the captions that appear in the edit-widget (a pair of tables) for a **Relationship** field of that type. (The captions cannot be edited.)

- The lower section specifies the set of fields to be displayed from each related issue record, along with their order and display widths.



## Working on a Relationship Types Subtab

1. In the upper section, click to select one of the relationship types.
2. In the lower section, click the *Setup Columns* button to bring up a dialog box in which you select a set of database fields, to be displayed as columns in a Relationship field's edit-widget. (See [The Setup Columns Command](#) on page 301.) You can also specify the order of the fields.
3. After closing this dialog box, you can adjust column widths and change the order of columns (see [Working with Tables](#) on page 9).

## The AccuWork Schema Editor

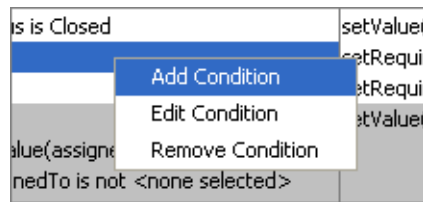
(Validation subtab)

Users of AccuWork create and modify issue records through the issue *edit form*. To increase the efficiency and accuracy of this process, you can create a set of validations to be applied as the user works with the edit form. (Validations are sometimes called "edit checks".)

You create and maintain the set of validations using a point-and-click interface on the Validation subtab. AccuWork displays the current validations in tabular format; each one takes the form "if a certain condition is true, then perform a particular set of actions".

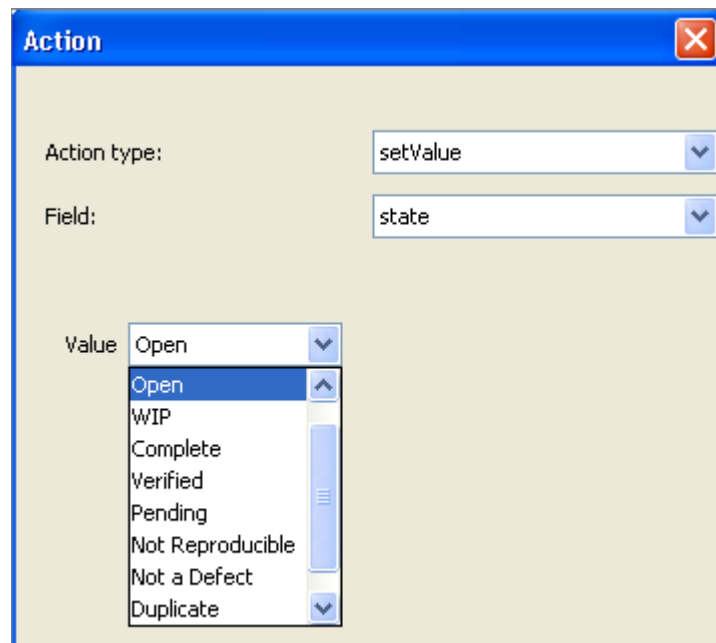
## Initializing Field Values in a New Issue Record

The first entry in the table of validations is special. Its condition is always "initialization of new issue", so that its actions are performed exactly once: between the time the user invokes the *New Issue* command and the time the new edit form appears.



To define an action, right-click in the *Actions* column, then select *Add Action* from the context menu. An *Action* window appears, in which you define the action. For initialization of a new issue record, the only appropriate action type is *setValue*. ( **Note:** Some releases of AccuWork allow you to include *setRequired* and *setChoices* actions in a record initialization. Such actions will be ignored when a new record is initialized.)

After setting the action type to *setValue*, select a field to be initialized and specify the initial value. The *Value* edit-widget adjusts to the selected field. In the example below, the *data type* of the *state* field is **choose**, so the list-box offers the field's predefined choices as the initial value.



## Conditional Validations

The setting of initial field values is an unconditional validation: it happens every time a *New Issue* command is invoked. All other validations are conditional: a certain set of actions are performed if, and only if, a certain condition is met.

The unconditional setting of initial field values occurs just once; but the conditional validations are performed repeatedly: when an edit form first appears *and* each time the user changes any field value. Each repetition involves:

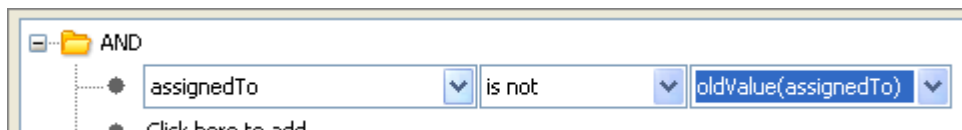
- Clearing the "required" status of all fields.
- Performing all validations (except for that first one: "initialization of new issue"). If a validation's condition is true, the corresponding actions are performed.

### Specifying the Condition

To create a new conditional validation -- that is, a new row in the table -- right-click anywhere in the Conditions column, then select *Add Condition* from the context menu. Then proceed to construct the "if" clause. The same context menu provides the *Edit Condition* command (for revising an existing "if" clause), and the *Remove Condition* command (for removing a conditional validation).

Specifying the "if clause" of a conditional validation is very much like specifying an AccuWork database query (see [Working in the Query Editor](#) on page 297). But there are some differences. In a validation condition, you can:

- Compare the current value of a field with its "old" value -- that is, compare the value currently displayed for the field vs. the value stored in the AccuWork database by the most recent *Save*.



- Compare the (current or old) value of a field with the (current or old) value of another field in the same record.

The first three screenshots show a validation rule with the operator 'is' and a folder icon. The first screenshot shows 'submittedBy' compared to 'assignedTo'. The second screenshot shows 'submittedBy' compared to 'oldValue(assignedTo)'. The third screenshot shows 'oldValue(submittedBy)' compared to 'closedBy'. The fourth screenshot shows 'oldValue(submittedBy)' compared to 'oldValue(closedBy)'.

- Test the AccuRev user identity or group membership of the person using the edit form (*CUR\_USER*).

The screenshot shows a validation rule with the operator 'is not' and a folder icon. The first field is 'CUR\_USER' and the second field is 'derek'. A dropdown menu is open, showing options: 'is', 'is not', 'is member of', and 'is not member of'.

Query conditions cannot make such field-to-field comparisons; they can only compare field values to literal values.

## Specifying the Actions

Each conditional validation can include any number of actions. You create an action by right-clicking in the Actions column of the validation, and selecting *Add Action* from the context menu. This displays a window in which you define the action.

After you've created one or more actions for a conditional validation, you can use the same context menu to revise or delete individual actions.

The following sections describe the actions that you can define to be performed if the validation condition is true. Each validation can invoke any number of actions, of any type.

- Setting a field value (*setValue*)
- Revising the choices for a "choose" field (*setChoices*)
- Requiring a value to be entered in a field (*setRequired*)

- Setting permissions on all or part of the issue record (*setIssuePermission*, *setTabPermission*, *setFieldPermission*)
- Requiring change set entries (*setChangesRequired*)
- Requiring a relationship with another issue record (*setParentRelationshipRequired*, *setChildRelationshipRequired*)

## Setting a Field Value

In addition to initializing field values (see Initializing Field Values in a New Issue Record), you can set the values of fields while the user is working with the record, based on certain conditions. For example, if the user enters *ColorStar* in the *program\_name* field, a validation could automatically set the *fix\_priority* field to *high*. (Management has mandated rapid improvement in the robustness of the ColorStar application.)

To define a *setValue* action for a validation condition:

1. In the Action window, select *setValue* as the action type, and select the field to be set.
2. The *Value* edit-widget adjusts to the selected field, just as it does when you're initializing a field in a new issue record.

For a "text", "log", or "timespan" field, you enter a text string as the value; for a "date" field, you specify a date in the standard way; for other field types, you use a list-box to specify any of the field's predefined values.

## Revising the Choices for a "choose" Field

The definition of each field of type "choose" includes an ordered list of strings. The user fills in this field by choosing one of the strings from a list-box.

In a validation, a *setChoices* action can change the "choice list" that the user will see when he opens the list-box. The changes to the choice list can include:

- Removing one or more of the existing choices
- Changing the order of the choices

You cannot add new strings to the choice list with a *setChoices* action. The changes made by a *setChoices* action are temporary: they last only until the next *setChoices* action on the same field, or until the user closes the edit form. When an edit form is opened on another issue record (or subsequently on this issue record), the user will see the field's original choice list, as defined on the *Schema* tab.

To define a *setChoices* action for a validation condition:

1. In the Action window, select *setChoices* as the action type, and use the *Field* list-box to select one of the database's fields of type "choose".
2. Use the controls at the bottom of the *Action* window to define the temporary change to the choice list:

## Requiring a Value to be Entered in a Field

Like many other programs that process fill-in-the-blanks forms, AccuWork can treat certain fields as required fields -- the user must specify a value in each such field before AccuWork will create

or update an issue record. (For a "choose", "list", or "user" field, the value must not be the *<none selected>* placeholder.)

AccuWork affords you great flexibility in controlling required fields. A field's "required" status is not part of the basic database schema. Instead, it is controlled by the conditional-validation facility. Thus, AccuWork repeatedly redecides whether a field is required: when the edit form first appears *and* each time the user changes any field value on the edit form. Whenever the user clicks the *Save* button, AccuWork uses the current set of required fields to allow or disallow the "save" operation.

To define a *setRequired* action for a validation condition:

1. In the Action window, select *setRequired* as the action type.
2. Select the field to be required.

If you want a field to be required in all circumstances, use the *setRequired* action along with a condition that's always true. For example, you can exploit the fact that every issue record has a positive integer as its issue number:

### Setting Permissions on All or Part of the Issue Record

The *setIssuePermission*, *setTabPermission*, and *setFieldPermission* actions are essentially similar: they restrict the editability of some component of the issue record:

- *setIssuePermission* makes the entire issue record read-only.
- *setTabPermission* makes a particular page (tab) of the issue edit form read-only.
- *setFieldPermission* makes a particular field of the issue record read-only.

There are two attributes that you can assign to these actions:

- *ro*— Make read-only for all situations.
- *auto*— (AccuWorkflow only.) Make read-only for end-users, but allow AccuRev to make changes.

The "read-only" ( *ro* ) permission affects the user's current access to the issue record; the "read-only" status is not stored in the repository as part of any issue record. For example, user *allison* might find that she cannot change any fields on the Assignment page of any issue record, because of this validation:

This restriction affect *allison* 's access to issue records; other users' access remains unaffected. And if this conditional validation is subsequently removed, *allison* will regain access to the Assignment page of all the issue records.

Note that setting read-only permission on an issue record will affect the behavior of AccuRev commands such as *cpkremove* and *promote*, in addition to restricting the user's ability to edit the issue.

The "automatic" ( *auto* ) permission is similar to *ro*, but in AccuWorkflow, AccuRev can perform actions that end-users are prevented from making as described above. For more information about AccuWorkflow-specific details, see "Modifying the AccuWork Schema" in the "Configuring AccuWorkflow" chapter of the *AccuRev Web Interface User's Guide*.

We recommend that each change to the schema be tested thoroughly to prevent unexpected side-effects to your organization's processes.

## Requiring Change Package Entries

The *setChangesRequired* action specifies that the user cannot *Save* an issue record unless there is at least one entry in the issue record's *change package*.

**Note:** be careful *not* to specify this action with the "initialization of new issue" condition (see [Initializing Field Values in a New Issue Record](#) on page 320).

## Requiring a Relationship with Another Issue Record

The *setParentRelationshipRequired* action specifies that the user cannot *Save* an issue record unless there is at least one entry in the lower table of a particular *relationship* field.

The *setChildRelationshipRequired* action specifies that the user cannot *Save* an issue record unless there is at least one entry in the upper table of a particular *relationship* field.

You cannot set both these requirements for the same *relationship* field, because each field can have parent relationships or child relationships, but not both.

## Revising and Removing Validations and Actions

Each conditional validation consists of a condition (left column) and a set of actions (right column). To revise or remove an entire validation, right-click anywhere within the condition, and select the appropriate command, *Edit Condition* or *Remove Condition* , from the context menu.

For the first (unconditional) validation, "initialization of new issue", you cannot remove or revise the condition. You can only work with the validation's actions.

To revise or remove an individual action from a validation, right-click on that action and select the appropriate command, *Edit Action* or *Remove Action* , from the context menu.

# The AccuWork Schema Editor

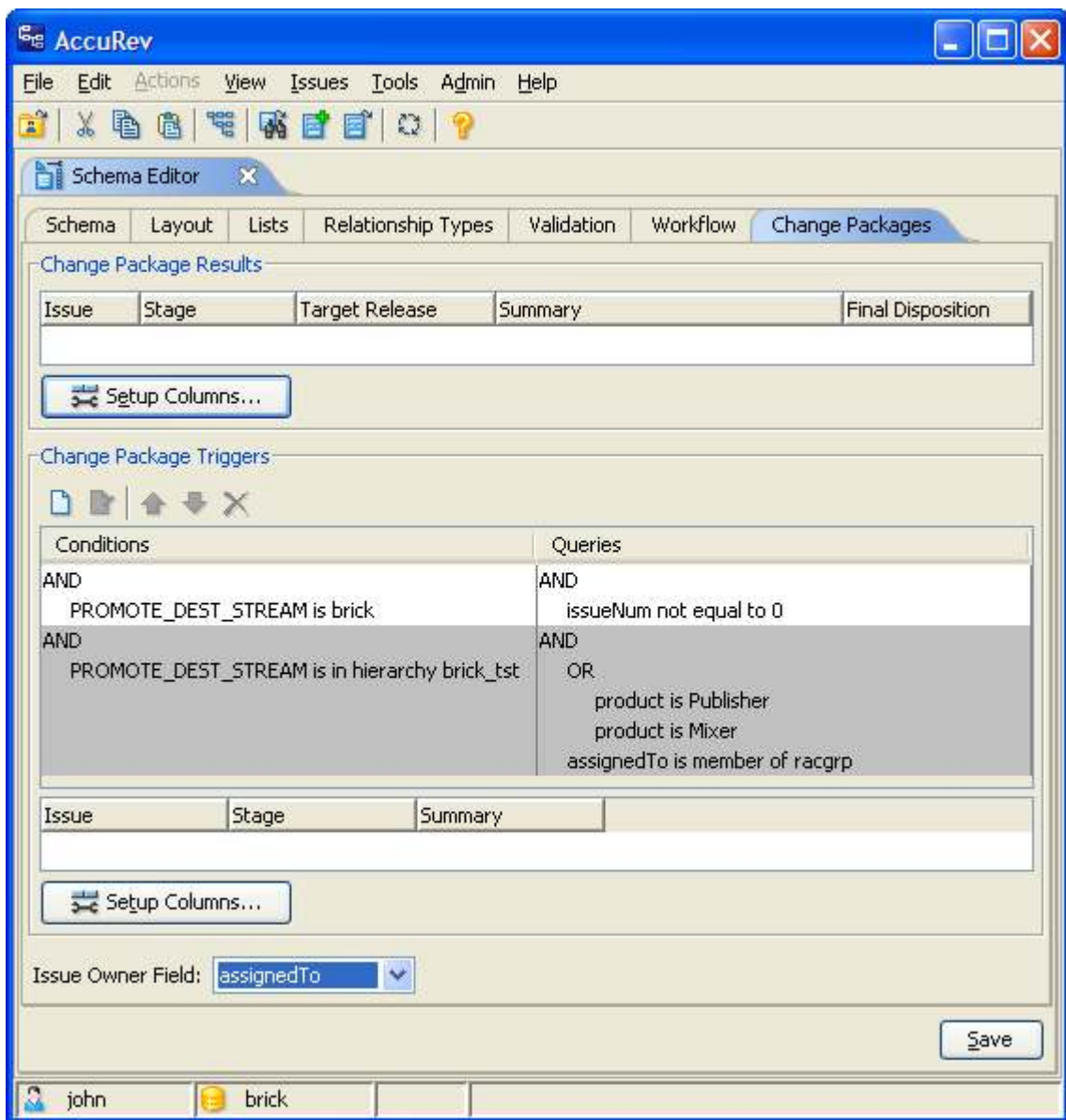
## (Change Packages subtab)

AccuRev Enterprise supports an enhanced integration between AccuRev and AccuWork: any AccuWork *issue record* can act as a *change package*, keeping track of which *versions* were created to fix the bug (or implement the new feature) described in that issue record. For a thorough discussion, see the topic [Change-Package-Level Integration between AccuRev and AccuWork](#) on page 328.

The Change Packages subtab configures the way in which change packages are reported and updated, on a depot-by-depot basis.

## Change Packages Subtab Layout

The Change Packages subtab specifies the change-package configuration for the current depot. It is organized as follows:



- In the *Change Package Results* section, you specify the format of the change-package table displayed by a *Show Issues* or *Show Diff By Issues* command. This set of fields is also used to identify issue records in the *Issue Dependencies* tab.
- In the *Change Package Triggers* section, you enable the change-package-level integration between AccuRev and AccuWork, by specifying one or more AccuWork queries, one of which will be invoked during execution of the *Promote* command. (The user chooses one or more of the issues records selected by the query; AccuRev adds the promoted versions to those issue record(s)' change packages.) In this section, you also specify the format of the table that displays the results of the *Promote*-time query to the user.
- In addition, the *Issue Owner Field* drop-down list displayed at the bottom of the tab can be set to any AccuWork field that contains user/group data. The field specified here displays

on the *Send to Issue* dialog, and allows the user to choose an issue belonging to another user or group.

## Working in the Change Package Results Section

Click the *Setup Columns* button to bring up a dialog box in which you select a set of issue fields, to be displayed as table columns. See [The Setup Columns Command](#) on page 301. You can also specify the order of the fields.

After closing this dialog box, you can adjust column widths and change the order of columns. (See [Working with Tables](#) on page 9.)

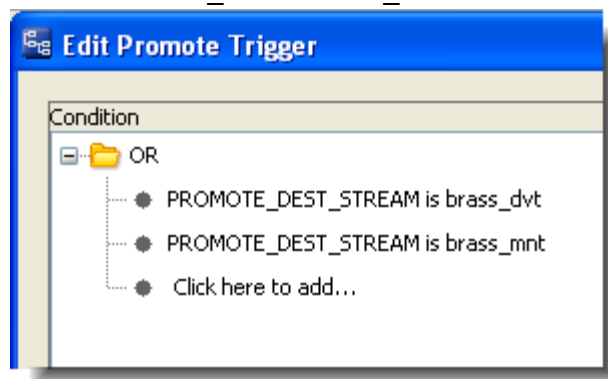
## Working in the Change Package Triggers Section

This section contains a set of condition/query pairs, one of which will be executed at *Promote*-time. If the first condition is satisfied, the first query will be executed; otherwise the second condition will be evaluated, and if it's satisfied, the second query will be executed; and so on. No error occurs if no condition is satisfied; the *Promote* command proceeds without triggering any change-package-level activity.

Note that there is no checkbox or button to enable the change-package-level integration. Rather, the existence of one or more condition/query pairs in this section implicitly enables the integration.

### Composing a Condition

Each clause of a condition performs a test on the *Promote* destination stream (indicated by the keyword *PROMOTE\_DEST\_STREAM*). For example, this condition is satisfied if the user is promoting to either of the streams **brass\_dvt** or **brass\_mnt**:



The mechanics of composing the condition are similar to that of composing an AccuWork query (see [Working in the Query Editor](#) on page 297).

### Composing a Query

Corresponding to each *Promote*-destination condition is any AccuWork query, to select a set of issue records. For the mechanics of composing the query, see [Working in the Query Editor](#) on page 297.

## Formatting the Table Displayed by the Change Package Trigger

To specify the format of the table displayed by the *Promote* command when it invokes the change-package-level integration between AccuRev and AccuWork, click the *Setup Columns* button to bring up a dialog box. Use this dialog box to select fields to appear as columns, the order of the columns (fields), and the sort order of the rows (records). After closing this dialog box, you can adjust the column widths and rearrange the columns.

## Change-Package-Level Integration between AccuRev and AccuWork

There are two similar, but separate facilities that integrate AccuRev's configuration management functionality with its issue management (AccuWork) functionality. The one described in this topic uses change packages as the point of integration. The other uses a particular issue-record field as the point of integration (see [Transaction-Level Integration between AccuRev and AccuWork](#) on page 331). Both integrations record information about the *Promote* transaction in a user-specified AccuWork issue record.

When a *Promote* command is executed a user's workspace (but not in a higher-level dynamic stream), the change-package-level integration records all the promoted versions on the *Changes* subtab of a user-specified AccuWork *issue record*.

## Enabling the Integration

The change-package-level integration is enabled on a depot-by-depot basis. Open the Schema Editor for a particular depot, and go to the Change Packages subtab. Filling in the lower section, "Change Package Triggers", enables the integration for that particular depot.

The Change Package Triggers section is structured as a set of condition/query pairs. One of the queries will be selected for execution at *Promote*-time. If the first condition is satisfied, the first

query will be executed; otherwise the second condition will be evaluated, and if it's satisfied, the second query will be executed; and so on.

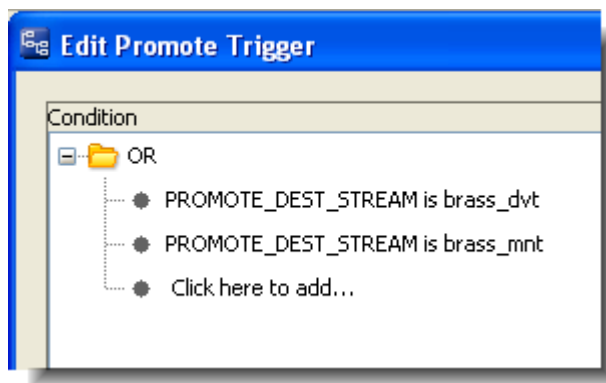
The screenshot shows the 'Schema Editor' window with the 'Change Packages' tab selected. It contains two main sections: 'Change Package Results' and 'Change Package Triggers'.

**Change Package Results:** This section has a table with columns 'Issue' and 'Short Description'. Below the table is a 'Setup Columns...' button. A red bracket on the right side of this section points to the text: "results format for Show Issues and Show Diff By Issues commands".

**Change Package Triggers:** This section contains a table with two columns: 'Conditions' and 'Queries'. The 'Conditions' column lists two AND conditions: "PROMOTE\_DEST\_STREAM is in hierarchy possum\_mnt" and "PROMOTE\_DEST\_STREAM is in hierarchy possum". The 'Queries' column lists two corresponding queries: "AND state is WIP type is defect" and "AND issueNum not equal to 0". Below this table is another 'Setup Columns...' button. A red bracket on the right side of this section points to the text: "one of these queries will be executed at Promote-time".

At the bottom of the 'Change Package Triggers' section, there is a table with columns 'Issue', 'Status', 'Priority', and 'Short Description'. A red bracket on the right side of this section points to the text: "and its result will be displayed in this format".

Each clause of a condition performs a test on the *Promote* destination stream. For example, this condition is satisfied if the user is promoting to either of the streams *brass\_dvt* or *brass\_mnt*:



The condition must test the destination stream (*PROMOTE\_DEST\_STREAM*) of a *Promote* command. The query corresponding to each condition can be any AccuWork query, which selects a set of issue records.

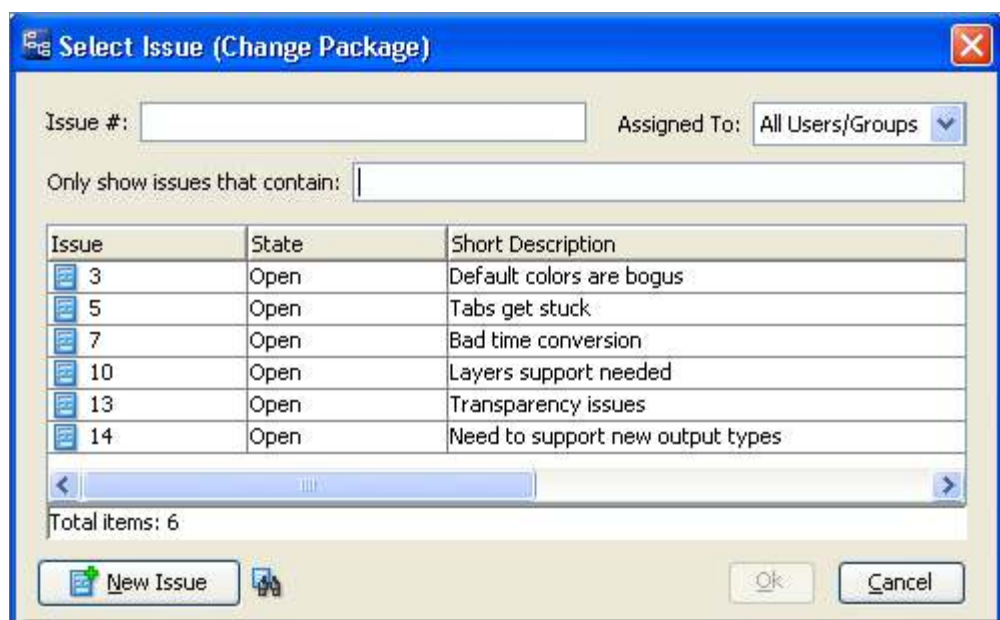
The Change Package Triggers section also specifies the format of each query's results table -- the fields to appear as columns, the column widths, the order of the columns (fields), and the sort order of the rows (records).

In addition, the *Issue Owner Field* drop-down list displayed at the bottom of the Change Packages tab can be set to any AccuWork field that contains user/group data. The field specified here displays on the *Send to Issue* dialog.

## Triggering the Integration

Once the integration is enabled for a depot, it is triggered whenever a user performs a *Promote* command in a workspace associated with that depot:

1. One of the AccuWork queries specified in the Change Package Triggers section is executed, selecting a certain set of issue records.
2. Those records are displayed to the user in the results table format specified in the Change Package Triggers section. In this example, AccuWork has been configured to allow the user to choose from issues assigned to another user or group if necessary.



3. The user selects one or more of the issue records, or types an issue identifier into the Issue # field.

**Promoting to Issues Other than Those Shown --** The dialog has a New Issue button, which enables the user to create a new issue record "on the fly". (Note: If you have AccuWorkflow enabled, and use a New Issue button from within a dialog in the Java client, you must set any AccuWorkflow fields in those issues manually. ) If the user hovers over the icon to the right of the New Issue button, it will display a tooltip containing the Change Package Triggers query. This makes it easier to create an issue that will be selected by the query, and thus available to promote against.

There may also be a user field to the right of the Issue # field that can be used to show issues belonging to another user.

**Typing an Identifier when using a Third-Party ITS --** If a field other than the default issueNum is specified as the 3pty ITS Key (on the Schema Editor's Schema subtab), specifying issue records by typing their identifiers can lead to errors: if multiple issue

records have the same identifier, AccuRev automatically adds the promoted version(s) to just one of those records — perhaps not the one you intended.

4. The command completes its work.
5. The versions involved in the command are recorded in the change package section (Changes page) of the selected issue record(s).

What happens if the user specifies no issue record or a non-existent issue record, such as 99999 or 0?

- If the user enters "0" (or equivalently, makes a blank or non-numeric entry), AccuRev checks whether issue record #0 is among the issues selected by the query executed Step 1. (Note: the query can select issue record #0, even though it doesn't exist -- for example with this clause: `issueNum equal to 0.`)
  - If the query does select issue record #0, the user's command completes but no information is recorded in the issue. This provides a way for the user to bypass the integration when performing the Promote command.
  - If the query does not select issue record #0, the user's command is cancelled, and no information is recorded in the issue.
- If the user specifies a non-existent issue record, such as "99999", the command is cancelled, and no information is recorded in the issue.

## What if Both Integrations are Enabled?

Both the change-package-level and transaction-level integrations can be enabled for a given depot at the same time. In this case, a user performing a *Promote* command in a workspace is prompted to specify an issue record just once, not twice. The prompting for an issue record by the change-package-level integration takes place as usual. That issue record is then updated by both integrations.

Note that even if both integrations are enabled, a *Promote* command performed in a dynamic stream (not a workspace) activates just the transaction-level integration, not the change-package-level integration.

## Transaction-Level Integration between AccuRev and AccuWork

There are two similar, but separate facilities that integrate AccuRev's configuration management functionality with its issue management (AccuWork) functionality. The one described in this topic a particular issue-record field as the point of integration. The other ([Change-Package-Level Integration between AccuRev and AccuWork](#) on page 328) uses change packages as the point of integration. Both integrations record information about the *Promote* transaction in a user-specified AccuWork issue record.

The integration between configuration management and issue management at the transaction level records the number of each *promote* transaction in a particular field of a user-specified issue record.

### Enabling the Integration

The transaction-level integration is enabled on a depot-by-depot basis, by setting the depot's *pre-promote-trig* trigger. For example:

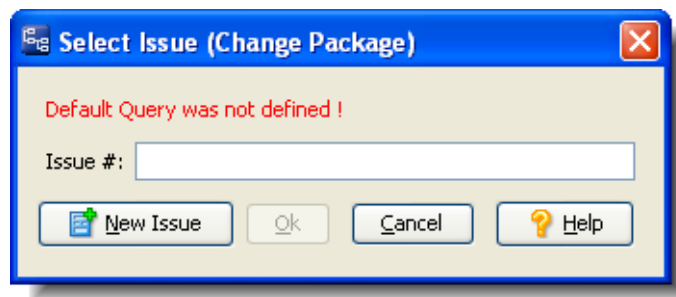
```
accurev mktrig -p kestrel pre-promote-trig client_dispatch_promote
```

Note that "client\_dispatch\_promote" is simply a keyword, not the name of a script file. The integration is implemented by two cooperating routines, one built into the AccuRev client software, one built into the AccuRev server software.

## Triggering the Integration

Once the integration is enabled for a depot, it is activated whenever a user executes the *Promote* command in any workspace or dynamic stream:

1. The depot's default query, as defined on the Queries tab (*Issues > Queries*), is executed and the results are displayed to the user.



2. The user selects one of the issue records. Note that if no default query is defined for the depot, the user is prompted to type an issue record number.
3. The *Promote* command completes its work, propagating the versions to the backing stream.
4. The *Promote* transaction number is recorded in the *affectedFiles* field of the selected issue record. (This change to the issue record is, itself, recorded as a transaction, of kind *dispatch*.)

If the user enters "0" or makes a blank entry, the *Promote* command completes but no change is made to any issue record. This provides a way for the user to bypass the integration.

Over time, the *affectedFiles* field of a given issue record can accumulate a SPACE-separated list of *Promote* transaction numbers.

## Implementation and Customization of the Transaction-Level Integration

When the transaction-level integration is activated, processing takes place on both the AccuRev client machine and the AccuRev server machine:

- The client-side processing -- querying the AccuWork issues database and prompting the user to specify an issue record -- is structured as a *pre-promote-trig* trigger routine built into the AccuRev client software.
- The server-side processing -- updating of the AccuWork issue record -- is structured as a *server-post-promote-trig* trigger routine built into the AccuRev server software.

You enable the integration by setting the *pre-promote-trig* trigger with the "client\_dispatch\_promote" keyword, as described above. You don't need to explicitly set a *server-post-promote-trig* trigger script. If you do, the script runs instead of -- not in addition to -- the server-side built-in routine.

In most cases, you'll want to avoid setting a *server-post-promote-trig* trigger script, just letting the built-in routines do their work. But suppose that after a *Promote*, you want the server machine to perform operations in addition to those defined in the transaction-level integration -- for example, updating reference trees and sending email messages. In such cases:

1. Create a script that performs the server-side part of the transaction-level integration, along with the desired additional processing. Start with the sample script *server\_dispatch\_promote\_custom.pl*, which is located in the *examples/dispatch* subdirectory of the AccuRev installation directory.
2. Place the script in the AccuRev *bin* directory.
3. Use a *mktrig* command to make the script the depot's *server-post-promote-trig* trigger script.

Further customizations of the transaction-level integration are possible. For example, you might want the user to be able to specify several issue records, not just one. Or you might want to link *promote* commands in one depot with the AccuWork issues database in another depot. Or you might want to update an issue record field other than *affectedFiles*. In such cases, you'll want to dispense with the built-in "client\_dispatch\_promote" routines altogether:

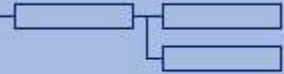
1. Start with the sample script *client\_dispatch\_promote\_custom.pl* (in the *examples/dispatch* subdirectory), and create your own script for use as a *pre-promote-trig* script to execute on the client.
2. As described above, start with the sample script *server\_dispatch\_promote\_custom.pl* (in the *examples/dispatch* subdirectory), and create your own script for use as a *server-post-promote-trig* script to execute on the server.

## If Both Integrations are Enabled

Both the change-package-level and transaction-level integrations can be enabled for a given depot at the same time. In this case, a user performing a *Promote* command in a workspace is prompted to specify an issue record just once, not twice. The prompting for an issue record by the change-package-level integration takes place as usual. That issue record is then updated by both integrations.

Note that even if both integrations are enabled, a *Promote* command performed in a dynamic stream (not a workspace) activates just the transaction-level integration, not the change-package-level integration.

## C. Support and Troubleshooting



### Customer Support Notes

To assist in a speedy resolution of problems with AccuRev usage or performance, AccuRev Customer Support personnel may ask you to provide data that you may not be accustomed to gathering on a day-to-day basis. The most frequently requested items are listed below, with instructions on how to gather the data. In your initial email communication with Customer Support, please include the data item(s) that you know (or suspect) are related to your specific issue.

#### Metadata collection

##### UNIX or Linux

1. Run the `accurev backup` command. This command will back up your **metadata** into the AccuRev `site_slice/backup` directory.
2. Use `accurev show slices` to find the location of the relevant depot directory (slice).
3. Stop the AccuRev Server.
4. tar your `site_slice` directory (including the backed-up metadata from step 1). Example:

```
cd /opt/accurev/storage/site_slice
tar -cvf ~/site_slice.tar . (don't forget the dot '.')
```

5. tar the relevant depot directory (slice), but exclude the data subtree. (The data subtree contains the content of the files you have placed into AccuRev source control; please do not send it to us!) Example:

```
cd /opt/accurev/storage/depots/ACME
tar -cvf ~/ACME.tar --exclude data . (the --exclude option is supported by GNU tar)
```

6. Compress the `.tar` files using `gzip`. Example:

```
gzip ~/site_slice.tar ~/ACME.tar
```
7. Start the AccuRev Server.
8. Send the gzipped files (`.tar.gz` suffix) to Customer Support. We recommend using FTP rather than email, since the files may be too large for your outgoing email server. Please contact `support@accurev.com` to obtain a username and password for our FTP server.

##### Windows

1. Run the `accurev backup` command. This command will back up your **metadata** into the AccuRev `site_slice/backup` directory.
2. Use `accurev show slices` to find the location of the relevant depot directory (slice).
3. Stop the AccuRev Server.

4. Create a ZIP archive of the entire site\_slice directory (including the backed-up metadata from step 1). Typically, it is C:\Program Files\AccuRev\storage\site\_slice.
5. Create a ZIP archive of the relevant depot directory (slice). Typically, it's a subdirectory of C:\Program Files\AccuRev\storage\depots. Remove the data subtree from the ZIP archive. (The data subtree contains the content of the files you have placed into AccuRev source control; please do not send it to us!)
6. Start the AccuRev Server.
7. Send the ZIP files to Customer Support. We recommend using FTP rather than email, since the files may be too large for your outgoing email server. Please contact support@accurev.com to obtain a username and password for our FTP server.

## Database inconsistencies

Gather the following information to begin troubleshooting the issue:

1. What changed in the environment? Power failure? Hardware failure?
2. What are the hardware specs on the AccuRev Server machine?
3. What is the AccuRev version on the AccuRev Server machine?
4. Get output of accurev info on the AccuRev Server machine.
5. Copy (ZIP or tar file) all AccuRev Server logs, located in subdirectory .../storage/site\_slice/logs. Search for any messages that might be the cause of your issue: disk failure, power outage, etc.
6. Copy all operating system logs (location varies from OS to OS).
7. Stop the AccuRev Server.
8. Get output of maintain dbcheck.
9. Start the AccuRev Server.
10. Gather AccuRev metadata.

## Running the AccuRev GUI in debug mode

If you experience your problem while running the AccuRev GUI, then running the GUI in Java debug mode can help diagnose and further troubleshoot your problem. Running in debug mode is also useful for monitoring the commands that the GUI sends to the AccuRev Server. This can help with scripting efforts, for example:

```
# Windows example - run the following commands in a Command Prompt window
cd C:\Program Files\AccuRev\bin
java -Daccurev.debug.acapi -cp "oro.jar;xercesImpl.jar;xml-apis.jar;fw.jar" fw.MainApp
```

```
# UNIX/Linux example - save and run the following shell script
#!/bin/sh
acbin=/opt/accurev/bin
acjre=/opt/accurev/jre/bin/java
cd $acbin
```

```
$accjre -Daccurev.debug.acapi -Daccurev.debug.env -classpath
"oro.jar:xercesImpl.jar:xml-apis.jar:fw.jar" fw.MainApp
```

Note that if you are running AccuRev 3.7 or earlier, the class path specified in the command should be "oro.jar, xerces.jar, fw.jar". If AccuRev is not installed in the standard location, you will also need to specify the correct path in your commands.

## User authentication problems

You can diagnose authentication problems with the *accurev secinfo* command:

```
accurev secinfo
```

The output is one of three keywords: anyuser, authuser, or notauth.

anyuser

The user is allowed to perform operations on streams except where permissions require an authenticated user.

authuser

The user is considered authenticated.

notauth

The user is not authenticated.

For more information on the *accurev secinfo* command, see the *AccuRev CLI User's Guide*.

## Getting AccuRev Server and client version information

How can you get the version information about the AccuRev Server without having to log into the server machine? You can perform either of the following operations on any AccuRev client machine:

- To get the integer version number of the AccuRev server and client software:

```
> accurev info -v
...
client_ver:    58
...
server_ver:    58
...
```

- To get the major/minor/patch version numbers of the AccuRev Server, use the *accurev xml* command:

1. Create a file (say, ver.xml), containing this very short XML document:

```
<serverInfo/>
```

2. Submit this file as an argument to the *xml* command:

```
accurev xml -l ver.xml
```

The command output in this format:

```
<serverInfo>
  <serverVersion
    major="4"
    minor="7"
```

```
    patch="2"/>
    <handleCount>10</handleCount>
  </serverInfo>
```

This example indicates that the AccuRev Server is Version 4.7.2.

## Integration issues

If you are reporting a problem with an AccuRev integration with a third-party IDE, send to Customer Support:

1. The name and version number of the IDE software.
2. The steps required to reproduce the problem. Be sure to include the type of project you're working with in the IDE and the location of the AccuRev workspace you're accessing through the IDE.
3. The version of AccuRev you are using.
4. (if applicable) The version of the AccuBridge integration you are using.

## AccuRev license issues

Send to Customer Support copies of:

- The *acserver.log* file (in *.../storage/site\_slice/logs*).
- The *keys.txt* file (in *.../storage/site\_slice*). [NOTE: keys.txt licensing is no longer supported as of AccuRev 5.4.]
- A screen shot of the error box or a copy of the error text.
- Output from *accurev info*, executed on the AccuRev Server machine.
- Output from *accurev show users*.

## Potential AccuRev defects

Send to Customer Support:

- A screen shot of the error dialog or a copy of the error text.
- The steps required to reproduce the problem. Include the depot's stream structure, if it is relevant.
- The operating system version on the client machine
- Are multiple users or multiple machines (client vs. server) involved?
- Output from these commands:

```
accurev
accurev info -v
accurev show wspaces
accurev show streams
accurev show slices
accurev show depots
```

`accurev diag`

## Replication problems

Send to Customer Support:

- A copy of the *acserver.cnf* file from both the master server and the replica server.
- A copy of the *acserver.log* file from both the master server and the replica server.
- Output from this command (if the problem is performance-related):

`accurev diag`

## Performance problems

Send to Customer Support the answers to this questionnaire:

1. Has performance always been poor? If not, what changed in the environment?
2. What exactly was the user trying to do? Does the performance problem occur with other users, on different client machines? Or is it a common issue on the same machine?
3. Does the operation actually finish? (A hang or lock-up can often be mistaken as performance problem.)
4. What are the AccuRev versions on the problem clients and server machines?
5. What are the hardware specs on the problem clients and server machines?
6. Is performance different in the AccuRev CLI and the AccuRev GUI?
7. Is the user's home directory on a remote disk? If so, consider moving the home directory so that it is on the machine where the AccuRev client software runs.

Send to Customer Support:

- Complete AccuRev server logs (in *.../storage/site\_slice/logs*)
- Two samples of output from *accurev diag*, run on the AccuRev Server machine in the depot storage directory.
- Two samples of output from *accurev diag*, run on the client machine in the user's workspace directory.
- Output from *accurev show triggers*.
- The contents of the trigger script files with the names listed below, located anywhere under the *storage* directory on the AccuRev Server machine:

`server_admin_trig`

`server_preop_trig`

The filenames might include a suffix, such as *.bat* on Windows.

- The AccuRev Server task list.

## Server task list

How do you create a listing of the AccuRev Servers currently active tasks (execution threads)?

1. Create a file (say, *tasks.xml*), containing this very short XML document:

```
<tasklist/>
```

2. Submit this file as an argument to the *xml* command:

```
accurev xml -l tasks.xml > tasklist.out
```

The contents of *tasklist.out* will resemble this example:

```
<tasklist>
  <task>
    <task_id>1299</task_id>
    <task_name>promote</task_name>
    <user>john</user>
    <host>192.168.89.222</host>
    <status>1</status>
    <starttime>2006/07/25 14:42:49</starttime>
    <seconds_on_server>2</seconds_on_server>
    <depot_write_lock>true</depot_write_lock>
    <progress_counter
      counter="0"
      count="6"
      max_count="11"/>
    <progress_counter
      counter="1"
      count="200"
      max_count="900"/>
  </task>
  <task>
    <task_id>1300</task_id>
    <task_name>xml</task_name>
    <user>john</user>
    <host>192.168.89.222</host>
    <status>1</status>
    <starttime>2006/07/25 14:42:51</starttime>
    <seconds_on_server>0</seconds_on_server>
  </task>
</tasklist>
```

## If you are not seeing an element...

If you cannot see an element that you believe that exists, there is a good chance that your AccuRev administrator has set element-level security using the [eac1](#) command, and you are being denied access to the element (or its container directory). For more information about EACLs and their behavior, see [Element ACLs \(EACLs\)](#) on page 256.

If you perform an Update, and the version displayed in the File Browser is correct but the content of the file does not appear to have been updated, this is typically due to a timestamp problem. It usually occurs when a file is changed outside of the workspace and then copied back in. The timestamp of the file that was dropped back into the workspace has a date and time stamp older than the date and time stamp of the file in the workspace. To update the file, temporarily uncheck [Timestamp Optimization](#) and perform the Update again. For more information, see [Validity of Traditional Timestamp Optimization](#) on page 100.

## The Logs Tab

The Logs tab provides access to command logs created during the current GUI session by these commands:

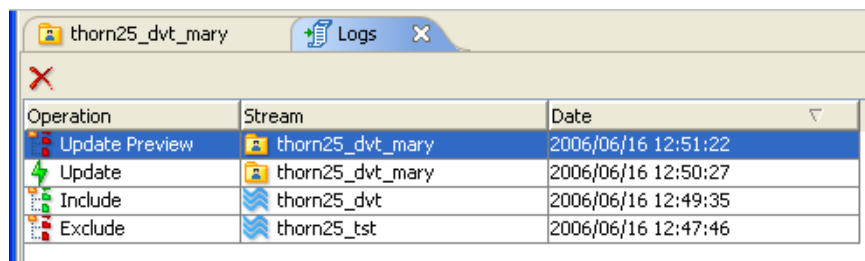
- Update and Update Preview
- Populate
- commands that change include/exclude rules

## Opening a Logs Tab

Choose *View > Logs* from the GUI main menu.

## Logs Tab Layout

The Logs tab contains a table with these columns:



Operation	Stream	Date
Update Preview	thorn25_dvt_mary	2006/06/16 12:51:22
Update	thorn25_dvt_mary	2006/06/16 12:50:27
Include	thorn25_dvt	2006/06/16 12:49:35
Exclude	thorn25_tst	2006/06/16 12:47:46

### Operation

The command that created the log: Update, Update Preview, Populate, Include, Exclude, Clear.

### Stream

The workspace or stream in which the operation took place.

### Date

A timestamp indicating when the operation took place.

## Working in a Logs Tab

You can display, save and delete individual logs, using these commands:

### View

Display the contents of the selected log in a text editor window. You can use environment variable `AC_EDITOR_GUI` or `EDITOR` to control which text editor gets invoked. If neither variable is set, AccuRev selects a program provided by the operating system.

### Save As

Copy the selected log to another location.

### Delete

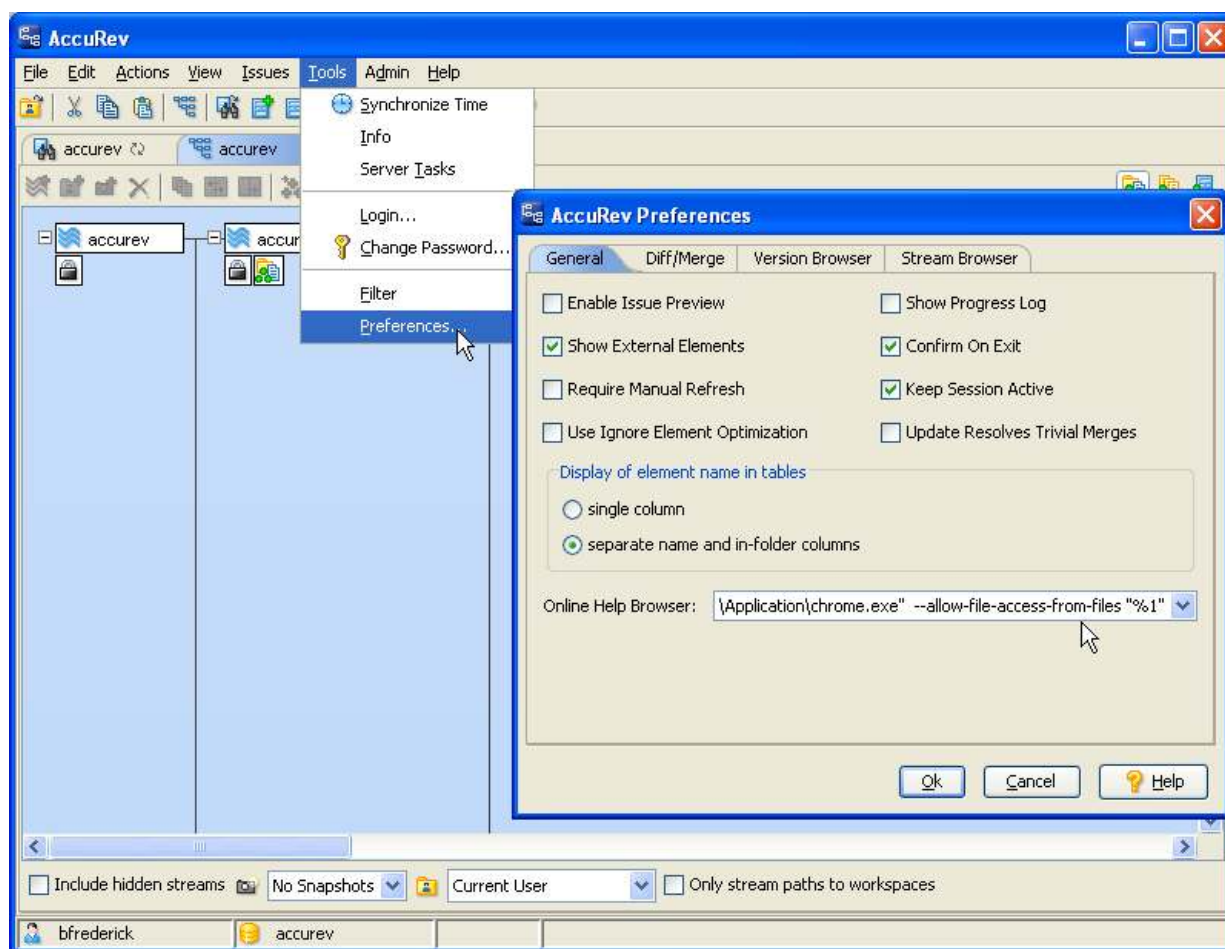
Remove the log file from disk storage (and from this table). Logs are stored in temporary files (for example, *C:\Documents and Settings\jjp\Local Settings\Temp\log3382.tmp*). These files are removed automatically at the end of the GUI session.

## Configuring a Web Browser to View Help Topics

The AccuRev help system is JavaScript-based to provide advanced search and navigation capabilities. Invoking help automatically launches a web browser, or opens a new tab in an existing web browser window, depending on which browser you use. Any JavaScript-enabled web browser is capable of displaying AccuRev GUI help topics, although you may need to adjust some security settings in more recent browser releases (see below).

By default, AccuRev tries to launch the operating system's default web browser. However, you can tell AccuRev which browser to launch by setting a preference that specifies the complete pathname of the web browser you wish to use. Each user can do this either in the GUI or through the Command Line Interface (CLI).

In the GUI, click Tools => Preferences and enter the full pathname of the browser in the Online Help Browser field on the General tab:



Or, use the *setpref* command from the CLI. For example:

```
accurev setpref AC_BROWSER "C:\Program Files\Mozilla Firefox\firefox.exe"
```

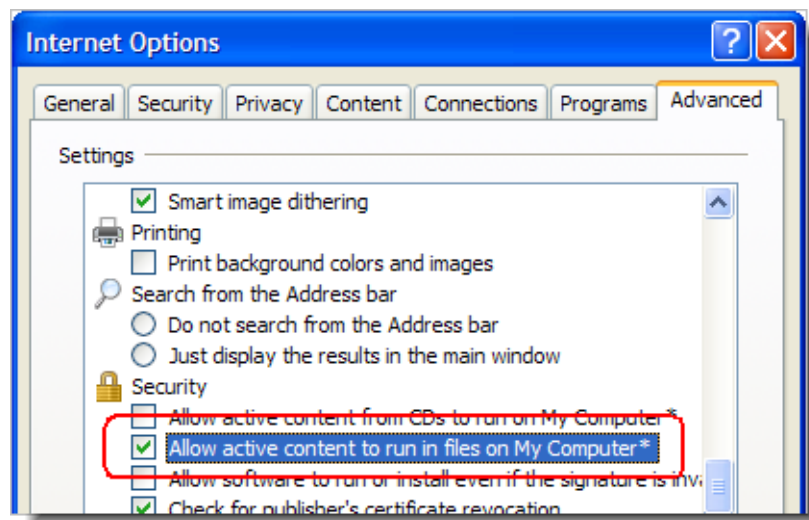
Either method updates the AC\_BROWSER setting in *preferences.xml* in the .accurev subdirectory of the user's home directory.

Note: If you are upgrading from AccuRev 4.9 and are having difficulty getting the Help to work, particularly on a 64-bit Windows system, check to see if you have an old browser preference set. If so, clear out this setting and let AccuRev use its default value to get going.

## Notes on Internet Explorer

### *Allowing "Active Content" with Internet Explorer 7*

The first time you use AccuRev Help with MS Internet Explorer, you may get a pop-up blocker. We have placed instructions on the default screen that appears with the pop-up blocker to guide you through the Advanced Settings of MSIE that will allow it to work with AccuRev help. These instructions are:



1. In MSIE, click *Tools > Internet Options*.
2. In the Internet Options window, click the Advanced tab.
3. Scroll down to the Security settings, and ensure that Allow active content to run in files on My Computer is checked.
4. Exit and restart MSIE.

## Notes on Google Chrome

Recent Chrome browsers have introduced a new security model which by default prevents them from working with Javascript-based web content residing on a local file system rather than on a server (see <http://code.google.com/p/chromium/issues/detail?id=47416>). This has caused several companies' help systems to stop working with these browsers.

Possible workarounds include:

- rolling back to an older version of Chrome
- using a different browser
- disabling the security feature in the AccuRev preferences
- disabling the security feature system-wide

Google has provided a command line switch to disable this new security feature. You can try disabling this feature just for AccuRev Help, by using the following procedure:

1. In the AccuRev GUI, go to Tools, Preferences
2. In the On-line Help Browser field, specify the start-up command line for Chrome with the "--allow-file-access-from-files" argument (Windows example shown):

```
"C:\Documents and Settings\<user>\Local Settings\Application  
Data\Google\Chrome\Application\chrome.exe" --allow-file-access-from-files "%1"
```

If this does not work for your installation, another option is to try using this switch as the default for all instance of Chrome on your system. The following example shows how to use this fix in Windows. Adjust appropriately for other operating systems. The following procedure assumes that you have Chrome set as your default .htm/.html browser:

1. Open Windows Explorer.
2. Navigate to Tools => Folder Options => File Types
3. Select "HTM" or "HTML", then click Advanced.
4. Select Open, then click Edit...
5. In the "Application used to perform action:" field, you should see a line similar to:

```
"C:\Documents and Settings\<user>\Local Settings\Application  
Data\Google\Chrome\Application\chrome.exe" -- "%1"
```

6. Replace the empty "--" with "--allow-file-access-from-files":

```
"C:\Documents and Settings\<user>\Local Settings\Application  
Data\Google\Chrome\Application\chrome.exe" --allow-file-access-from-files "%1"
```

7. Click OK, OK, and Close to save your changes and exit out of the dialogs.

Of course, if this line has already been customized at your site, you will need to adjust the above instructions as needed.

Also, note that implementing this fix will disable a security change introduced with Chrome, so the security policies of some organizations may prohibit them from doing this.

# Index

## A

- accelerators
  - keyboard 7
- Access Control List, element 256
- Access Denied 256
- AccuRev Web UI 7
- ACCUREV\_IGNORE\_ELEMS 94, 95, 96, 101
- AccuWork
  - export file types 14
- AccuWorkflow 8
- ACLs 256
- Add to Depot command 108
- Alt key 7
- Anchor 81
- Anchor command 110
- Annotate tab 111

## B

- backed status 73, 91
- browsable data structures 58

## C

- Change Package 130
- Change Password command 37
- Changing a Directory 74
- characters 14
  - allowed 14
  - restrictions 14
- Choose Depot 12
- Choose Stream dialog 13
- Clone Tabs 6
- columns
  - table 9
- Confirm On Exit 43
- contacting technical support iv
- context-sensitive help 1
- Conventions iv
- conventions
  - filename 14
- Copy URL to Clipboard 8
- corrupted status 89

## D

- Deep Overlap
  - optimization 64
- Deep Overlap Optimization 64
- Default Group 134
- Defunct command 113
- defunct status 89
- Delete command 115
- Deleting a File 75
- Depot Name 16
- depot-relative pathname 88
- Depots
  - choosing 12
- depots
  - case-sensitivity 16
  - physical location 16
- Depots tab 47
- Details Pane 66, 79
- Diff/Merge 44
- Display of element name in tables 44

## E

- EACLs 256
- Editing a File's Contents 70
- Element Status 88
- Element Types 136
- elink status 89
- Enable Issue Preview 42
- Esc key 7
- excluded status 90
- exclusive file locking 17
- external objects
  - display preference 101
- external status 90

## F

- F5 key 7
- File Browser 55
  - alternatives to 57
- File Chooser dialog 13
- file locking 17
- File Menu 16
- File Status Indicators 89
- filename
  - conventions 14

- restrictions 14
- filter 9
  - users or groups 40
- Filter Results 12
  - for Find Stream 154
  - for Search by Element Name 65
- Filter Streams command 37
- find 9
  - stream 153
- Folders pane 58

## G

- groups
  - filter 40
- GUI
  - overview 5
  - starting 5
- GUI Window Indicators 56

## H

- Help 1
- help
  - accessing without client 3
  - browser preference 44
  - configuring browser for 4
- How\_configure 4

## I

- Ignore Changes in Whitespace 45
- Ignore Whitespace 45
- Include/Exclude Mode 59, 101

## K

- Keep command 115
- Keep Session Active 44
- kept status 73, 91
- keyboard accelerators 7

## L

- lengths, character and pathnames 14
- locking
  - file 17
- Locks command 53
- Login command 35

- Login Dialog 35

## M

- member status 73, 91
- Merge 46
- merge 77, 78
- Merge vs. Patch 119
- missing status 90
- missing-target status 89
- modified status 73, 90
- modified-target status 89

## N

- name
  - display preference 88
- name length 14
- naming conventions 14
- New Depot command 16
- New File command 118
- New Folder command 117
- New Reference Tree command 51
- New Workspace command 18
- no such elem 256
- no such elem status 90

## O

- on-line help 1
- Online Help Browser 44
- Open in Web 8
- Open Issues in Web Only 42
- Open Workspace dialog 18
- optimization
  - Deep Overlap 64
  - pathname 94
  - search 64
  - timestamp 96
- overlap status 79, 91, 122

## P

- Patch algorithm 121
- Patch From command 119
- Patch vs. Merge 119
- pathname lengths 14
- Pathname Optimization 94
- PDF 2

- accessing AccuRev manuals 3
- Populate command 121
- preference
  - display name 88
- preferences
  - display of external objects 101
- Preferences Command 42
- progress box 122
- Promote Command 123
- Promote command 72, 122
- Properties command 32

## R

- Real Versions and Virtual Versions 136
- real versions and virtual versions 122
- Reference Trees tab 49
- Refresh 7
- Rename command 124
- Renaming or Moving a File 71
- Require Manual Refresh 43
- resizing table columns 9
- restrictions
  - character 14
  - filename 14
  - naming 14
- Revert by Change Package 161
- Revert Change Package 161
  - using a workspace 161
  - Without Using a Workspace 162
- Revert command 126
- Revert to Backed 73
- Revert to Backed command 127
- Revert to Most Recent Version command 128
- rows
  - sorting 10
- Rules Pane 103

## S

- scan threshold 97
- search 9
  - by element name 65
  - by status 59
  - by stream name 153
  - stream 153
- searches 60

- Searches Pane 59
- Send to Issue (specifying basis) 128
- Send to Issue command 128
- Send to Workspace command 131
- Server Tasks tab 34
- Show External Elements 43
- Show Info dialog 33
- Show Progress Log 42
- Slices tab 51
- slink status 89
- snapshot
  - New Snapshot command, 28
- stale status 91
- stranded status 90
- stream
  - find 153
  - search 153
- Stream Filter 38
- StreamBrowser
  - preferences 47
- streams
  - adding 39
  - choosing 13
  - filtering 40
  - removing 40
  - searching 40
- support
  - contacting technical support iv
- Synchronize Time command 33

## T

- table
  - rows 10
- table columns 9
- tables 9
- tabs
  - cloning 6
  - closing 6
  - GUI 5
  - refreshing 6
- technical support
  - contacting iv
- tech\_support iv
- threshold 31
- Timestamp Optimization 96
- timestamp optimization 31, 96

- toolbar 1
- trigger 127
- Triggers tab 51
- twin status 90
- typographical conventions iv

## U

- underlap status 79, 91, 122
- update 77
- Update command 132
- Update Resolves Trivial Merges 44
- Use Ignore Element Optimization 44
- users
  - filter 40

## V

- Version Browser
  - preferences 47
- Virtual Versions and Real Versions 136

## W

- Web UI, AccuRev 7
- WIP (Work in Progress) tab 133
- workspace
  - choosing for Revert command 127
  - new 18
  - open dialog 18
- Workspaces tab 25

## X

- xlinked status 90