AccuRev

AccuBridge for IntelliJ IDEA

User’s Guide

Version 2010.1

May, 2010

20-May-2010

AccuBridge for IntelliJ IDEA User's Guide

Table of Contents

Introduction

This document describes ***AccuBridge^{TM} for IntelliJ$^{®}$ IDEA***, the integration of AccuRev with JetBrains's **IntelliJ IDEA** Integrated Development Environment (IDE). The Integration is implemented as a standard plug-in, enabling IDE users to access AccuRev version-control facilities using IntelliJ IDEA's own menu structure.

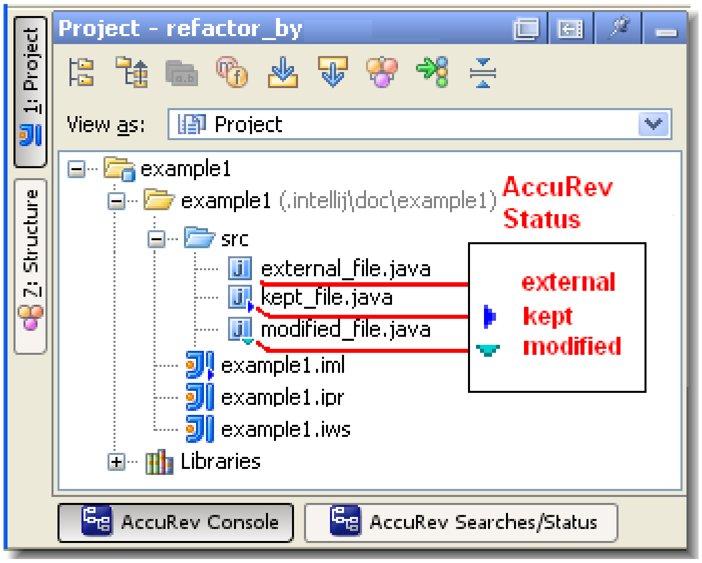
For updated information on platform support, visit the AccuBridge download page on the AccuRev, Inc. Web site.

AccuRev Enhancements to the IDE

AccuBridge for IntelliJ IDEA extends the IDE in many ways, as described in the sections below.

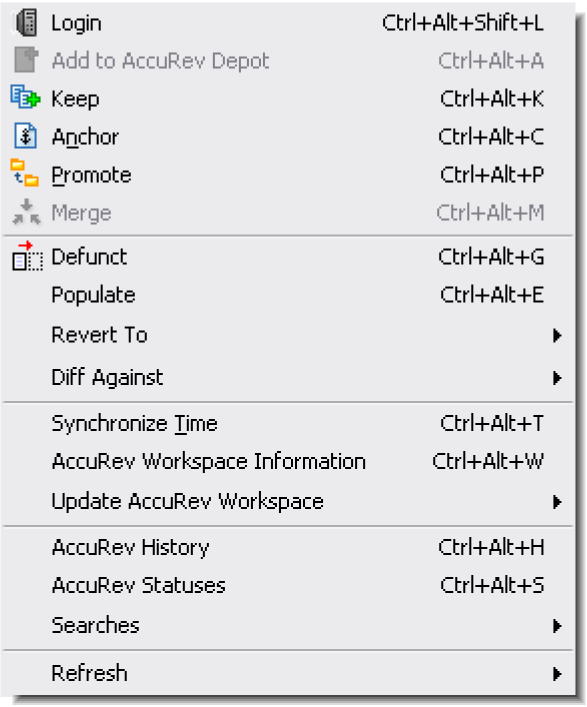
AccuRev Status Indicators

Files managed by AccuRev acquire icon decorations in the Project tool subwindow list showing their AccuRev status, as shown. Icons for directories are not decorated, even though they are under AccuRev version control, as well.



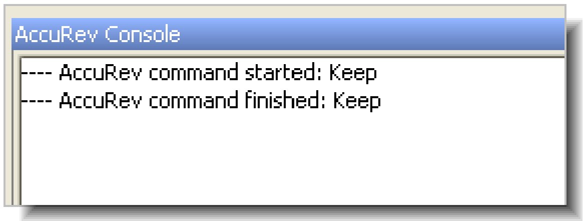
AccuRev Commands

You can invoke a substantial number of AccuRev's version-control commands from context (right-click) menus and from the **Version Control** > **AccuRev** menu. For a complete reference to the commands available through the Integration, see *Executing AccuRev Commands* on page 13.

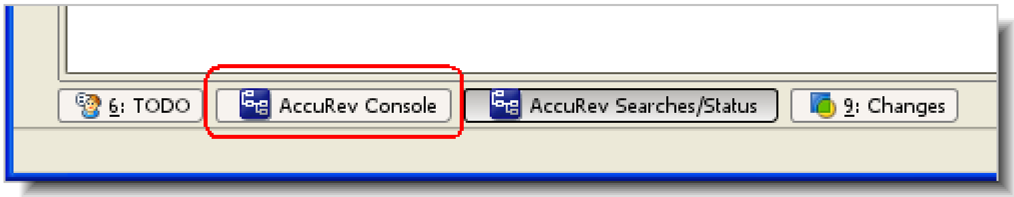


AccuRev Console Tool Subwindow

Many AccuRev commands generate messages, which are displayed in the AccuRev Console tool subwindow. This subwindow is hidden by default. To open or close it, click its button at the bottom of the IDE window, or invoke the command **Window** > **Tool Windows** > **AccuRev Console**.

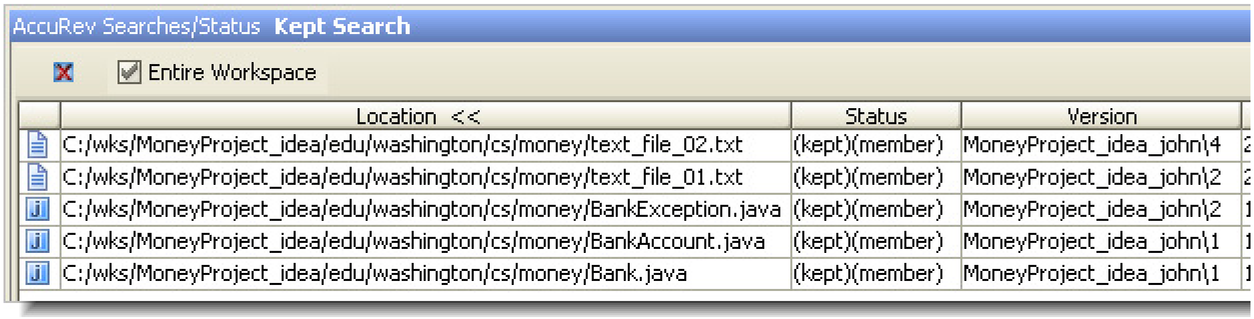


```
---- AccuRev command started: Keep
---- AccuRev command finished: Keep
```



AccuRev Searches/Status Tool Subwindow

The **AccuRev Statuses** command and the various forms of the **AccuRev Searches** command display their results in the AccuRev Searches/Status tool subwindow:

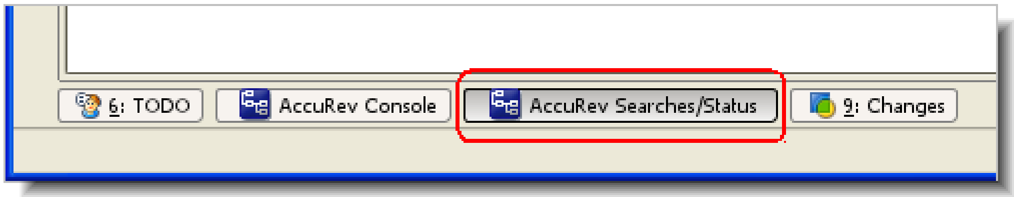


AccuRev Searches/Status **Kept Search**

☑ Entire Workspace

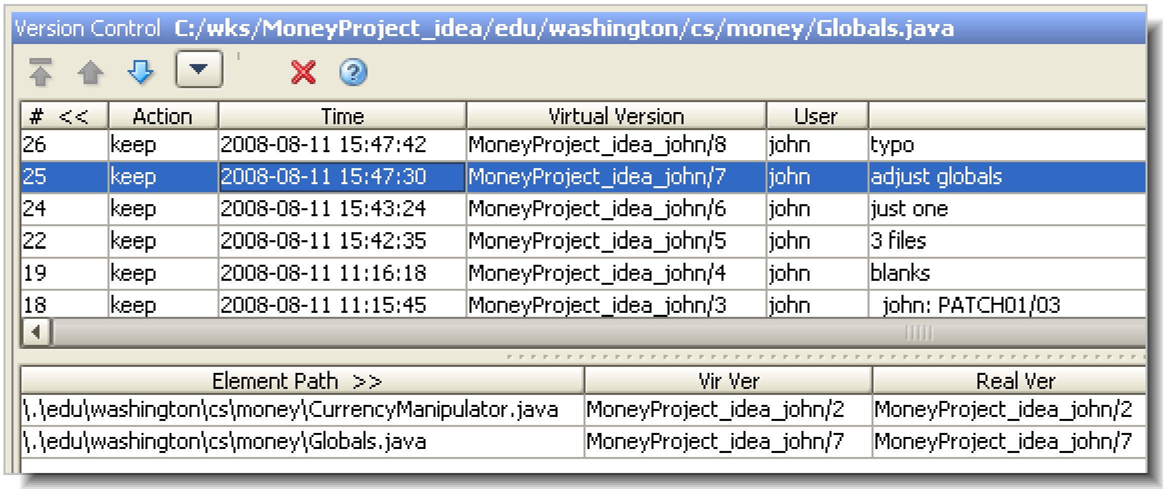
Location <<	Status	Version	
C:/wks/MoneyProject_idea/edu/washington/cs/money/text_file_02.txt	(kept)(member)	MoneyProject_idea_john\4	2
C:/wks/MoneyProject_idea/edu/washington/cs/money/text_file_01.txt	(kept)(member)	MoneyProject_idea_john\2	2
C:/wks/MoneyProject_idea/edu/washington/cs/money/BankException.java	(kept)(member)	MoneyProject_idea_john\2	1
C:/wks/MoneyProject_idea/edu/washington/cs/money/BankAccount.java	(kept)(member)	MoneyProject_idea_john\1	1
C:/wks/MoneyProject_idea/edu/washington/cs/money/Bank.java	(kept)(member)	MoneyProject_idea_john\1	1

To open and close this subwindow, click its button at the bottom of the IDE window, or invoke the command **Window > Tool Windows > AccuRev Searches/Status**.



AccuRev Element History

The **AccuRev History** command displays the set of transactions involving a selected element in the Version Control tool subwindow.



Version Control **C:/wks/MoneyProject_idea/edu/washington/cs/money/Globals.java**

# <<	Action	Time	Virtual Version	User	
26	keep	2008-08-11 15:47:42	MoneyProject_idea_john/8	john	typo
25	keep	2008-08-11 15:47:30	MoneyProject_idea_john/7	john	adjust globals
24	keep	2008-08-11 15:43:24	MoneyProject_idea_john/6	john	just one
22	keep	2008-08-11 15:42:35	MoneyProject_idea_john/5	john	3 files
19	keep	2008-08-11 11:16:18	MoneyProject_idea_john/4	john	blanks
18	keep	2008-08-11 11:15:45	MoneyProject_idea_john/3	john	john: PATCH01/03

Element Path >>	Vir Ver	Real Ver
\.\edu\washington\cs\money\CurrencyManipulator.java	MoneyProject_idea_john/2	MoneyProject_idea_john/2
\.\edu\washington\cs\money\Globals.java	MoneyProject_idea_john/7	MoneyProject_idea_john/7

Checkout from Version Control for AccuRev

The command **Version Control > Checkout from Version Control > AccuRev** creates a new AccuRev workspace, populating it with files already under version control in a particular AccuRev depot. Then, it makes the workspace into an IntelliJ IDEA project.

Getting Started with AccuBridge for IntelliJ IDEA

This chapter describes tasks that get you up and running with AccuBridge for IntelliJ IDEA.

Establishing Your Identity

All AccuRev commands must be executed by an AccuRev user. To use the Integration, you must establish your identity to AccuRev. AccuRev has two schemes for authenticating users:

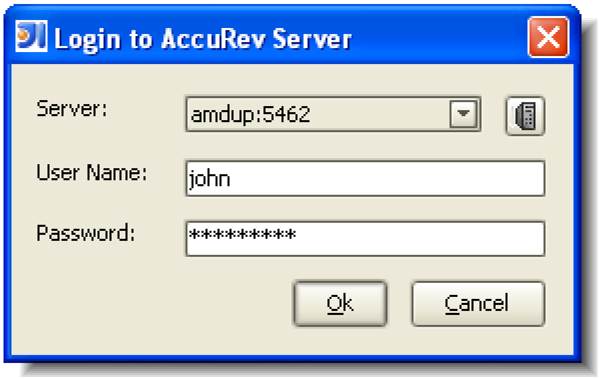
- With the *traditional* user-authentication scheme, AccuRev defaults to using your operating-system username as your AccuRev username. Your AccuRev password must be stored in file **authn**, in the **.accurev** subdirectory of your operating-system home directory. (On Windows, this is %HOMEDRIVE%\%HOMEPATH%.)

  To customize this behavior, set either or both of the following environment variables before starting the IDE (Windows users should set user-level, not system-level variables):

  - To use a different username, set ACCUREV_PRINCIPAL to that username.
  - To place your **.accurev** directory elsewhere, set ACCUREV_HOME to the new location.

- With the *accurev_login* user-authentication scheme, you must perform an explicit login at the AccuRev level. You can do this before starting IntelliJ IDEA. When you create a workspace in the IDE, you are prompted to log in if you aren't already logged in. When working within an IDE project, you can use the **AccuRev** > **Login** command to log in. It's available on context menus in the Project tool subwindow or under **Version Control** on the main menu.

  In the Login dialog box, select an AccuRev Server, and enter a username/password pair. Then click **Ok**.


Login to AccuRev Server

Server: amdup:5462

User Name: john

Password: **********

See the *AccuRev Administrator's Guide* for complete information on AccuRev user management.

Placing Projects Under AccuRev Version Control

The Integration provides access to AccuRev version control commands within IntelliJ IDEA. The commands executed by the Integration move data between a central source-code repository (a depot) and a personal AccuRev work area (a workspace). AccuRev can manage an IDE project's files if (and only if) the project's location on disk is within an AccuRev workspace. It doesn't matter which is created first — the IDE project or the AccuRev workspace.

Also relevant is this aspect of AccuRev's data architecture: AccuRev does not automatically version-control every file within an AccuRev workspace. For example, version-controlling text-editor backup files would be a waste of resources. Files created in a workspace initially have **(external)** status. They become version-controlled elements when explicitly processed with the **Add to AccuRev Depot** command.

Given these facts, it makes sense to consider the following use cases. Before performing any of the procedures, make sure that AccuRev depots and streams have been created and properly configured. These operations cannot be performed within IntelliJ IDEA.

- **Case 1:** An IntelliJ IDEA project already exists; you want to make AccuRev its version control provider.

  Perform the procedure in section *Making AccuRev the Version Control Provider for an Existing IDE Project* on page 7.

- **Case 2:** Your sources are already under AccuRev version control; you want to use IntelliJ IDEA for development.

  If you have access to an AccuRev workspace containing the sources, perform the procedure in *Creating a New IDE Project in a Workspace* on page 10.

  If you need to create a new AccuRev workspace containing the sources, perform the procedure in *Creating a New Workspace for Sources that are Already Under AccuRev Version Control* on page 8. With IntelliJ IDEA 6, you must also perform the procedure in *Creating a New IDE Project in a Workspace* on page 10.

- **Case 3:** Your sources are located in an AccuRev workspace; an IntelliJ IDEA project exists for the sources.

  This is your goal! Invoke the IDE's **File > Open Project** command, navigate to the project file (**.ipr**), and click **OK**.

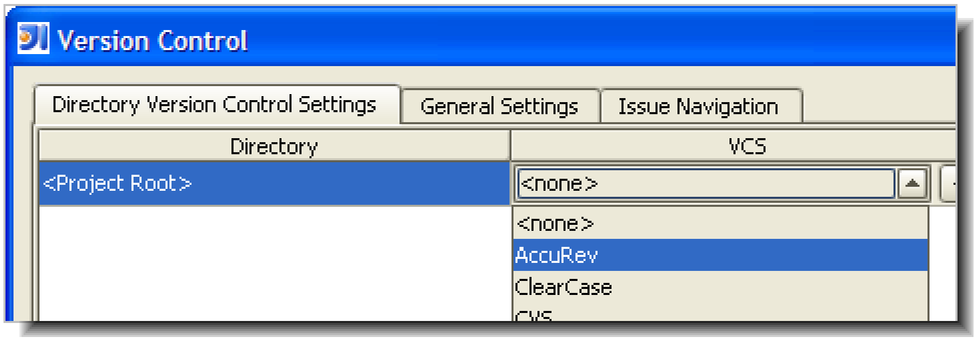
The remainder of this chapter contains the sections referenced above.

Making AccuRev the Default Version Control Provider for New IDE Projects
Making AccuRev the Version Control Provider for an Existing IDE Project
Creating a New Workspace for Sources that are Already Under AccuRev Version Control
Creating a New IDE Project in a Workspace
Converting (external)-Status Files to Version-Controlled Elements

Making AccuRev the Default Version Control Provider for New IDE Projects

After you perform this procedure, AccuRev version control will be enabled automatically for any project created with the **New Project** command, as long as the specified “project file location” for the new project is within an existing AccuRev workspace.

1. Close any open projects.
2. Select **File > Settings** from the IntelliJ IDEA main menu.
3. If the button in the bottom left corner of the Settings dialog reads **Default View**, click it.
4. Click **Version Control**.
5. On the Directory Version Control Settings tab, in the Directory entry for <*Project Root*>, choose **AccuRev** from the drop-down VCS list.



6. Click **OK**, then **Close** to dismiss all the pop-up windows.

Making AccuRev the Version Control Provider for an Existing IDE Project

You might want to use AccuRev selectively — for some new projects but not others. Or you might have existing projects to be used with AccuRev. This procedure handles both of these cases:

1. Arrange for the project files to be located within an existing AccuRev workspace, in either of these ways:

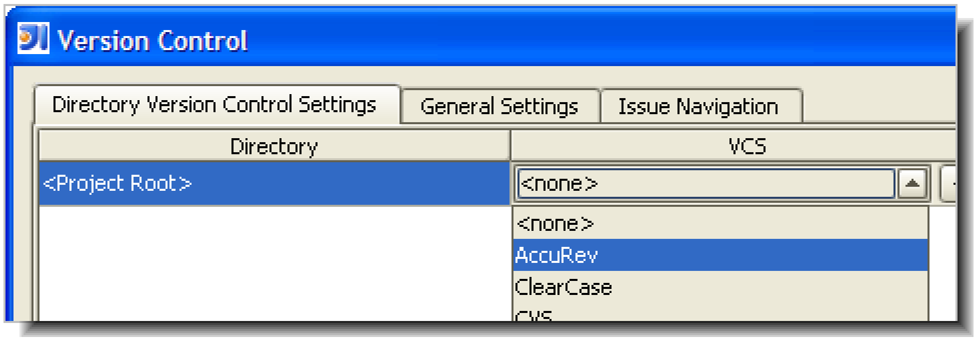
   - Move all the project files from their current location to a location within an existing AccuRev workspace.
   - Use the AccuRev GUI to convert the project’s current disk location, or a higher-level directory, into a new AccuRev workspace (**File > New > Workspace** command).

   In either case, the project’s root directory need not be the top-level directory of the workspace.

2. Open the project in IntelliJ IDEA.
3. Select **File > Settings** from the IntelliJ IDEA main menu.
4. If the button in the bottom left corner of the Settings dialog reads **Default View**, click it.

5. Click **Version Control**.

6. On the Directory Version Control Settings tab, in the Directory entry for <*Project Root*>, choose **AccuRev** from the drop-down VCS list.



7. Click **OK**, then **Close** to dismiss all the pop-up windows.

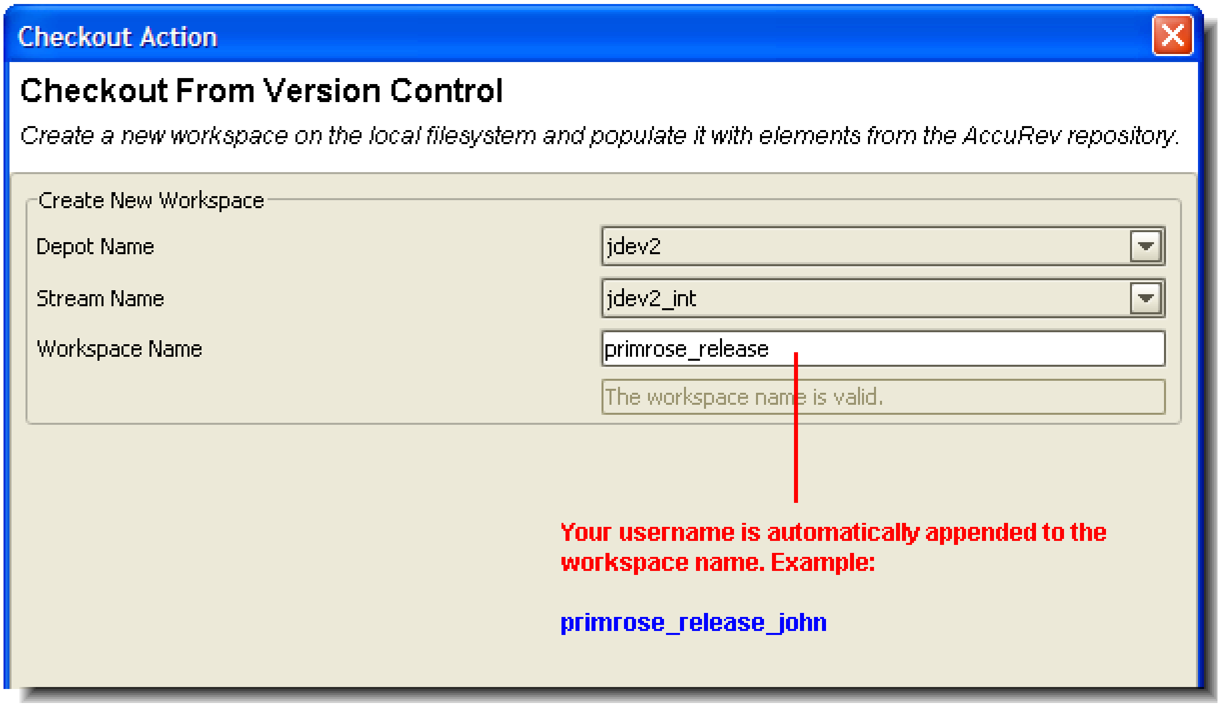
8. If you moved data into an existing workspace or created a new workspace, as outlined in Step 1, you must place the project's files under version control. See *Converting (external)-Status Files to Version-Controlled Elements* on page 11.

Creating a New Workspace for Sources that are Already Under AccuRev Version Control

Use this procedure if someone has already placed a set of development files under AccuRev version control, on his or her own machine. This procedure creates a new workspace for your personal use, and "updates" the new workspace with the latest versions of the development files.

1. Invoke the command **Version Control > Checkout from Version Control > AccuRev**. (If you are using the *accurev_login* authentication mode, you might be required to log in at this time.)

2. On the first wizard screen, specify an AccuRev depot, a stream within that depot, and a name for the new workspace. AccuRev automatically adds your username as a suffix to the name you enter (unless you type the suffix yourself).



Checkout Action

Checkout From Version Control

Create a new workspace on the local filesystem and populate it with elements from the AccuRev repository.

Create New Workspace

Depot Name: jdev2

Stream Name: jdev2_int

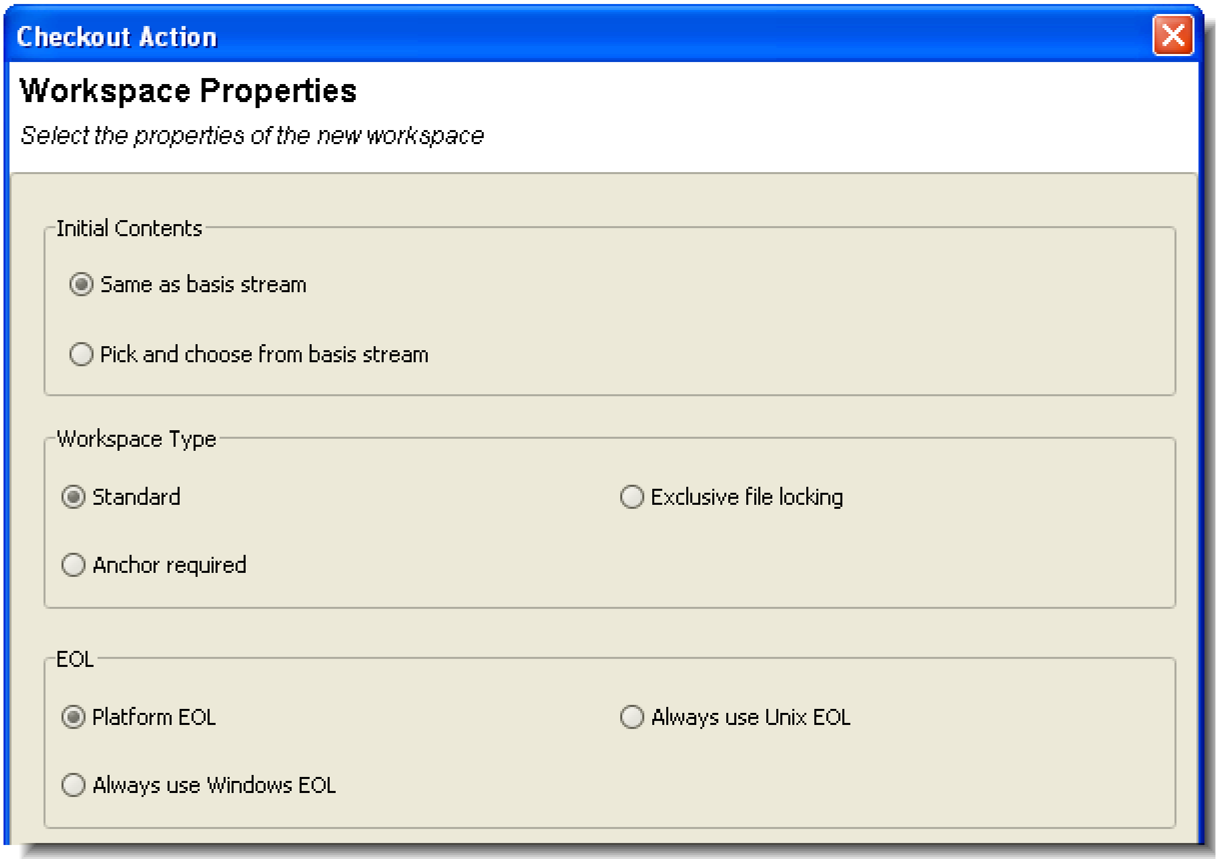
Workspace Name: primrose_release

The workspace name is valid.

Your username is automatically appended to the workspace name. Example:

primrose_release_john

3. On the second wizard screen, accepting the defaults is often most appropriate. See *Anchor* on page 14 for a discussion of the exclusive file locking and anchor required features.



Checkout Action

Workspace Properties

Select the properties of the new workspace

Initial Contents

● Same as basis stream

○ Pick and choose from basis stream

Workspace Type

● Standard ○ Exclusive file locking

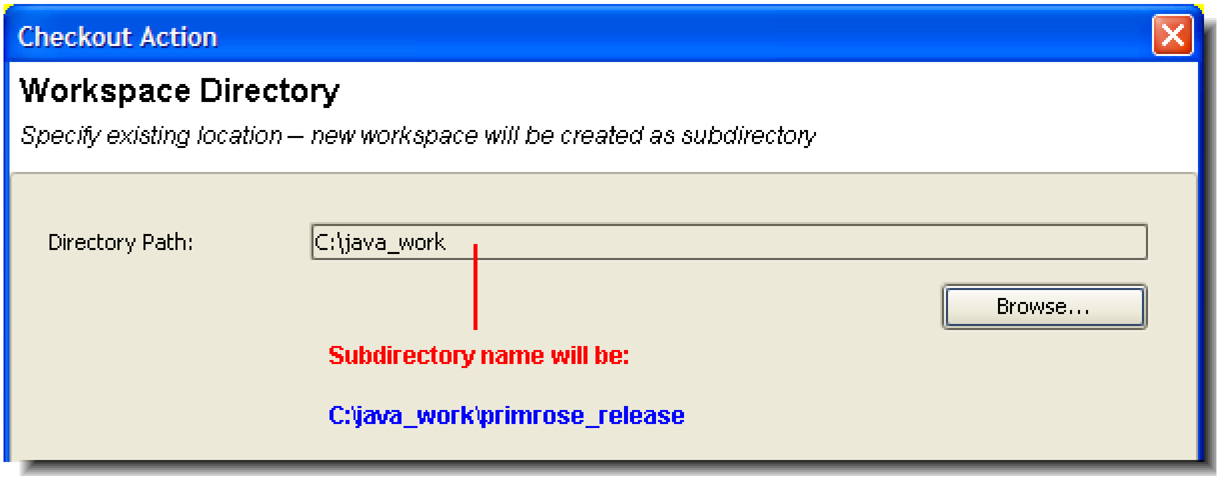
○ Anchor required

EOL

● Platform EOL ○ Always use Unix EOL

○ Always use Windows EOL

4. On the third wizard screen, click the **Browse** button and navigate to an existing directory.



Checkout Action

Workspace Directory

Specify existing location – new workspace will be created as subdirectory

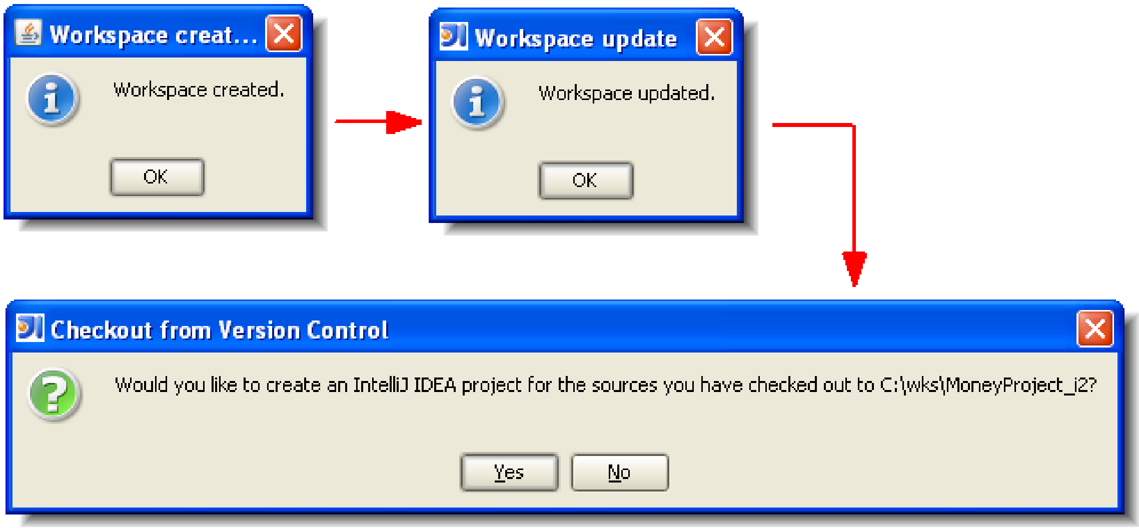
Directory Path: C:\java_work

Subdirectory name will be:

C:\java_work\primrose_release

The new workspace will be created as a subdirectory at the location you specify. The subdirectory name is the same as the workspace name you specified on the first wizard screen.

5. IntelliJ IDEA 7 automatically proceeds to create an IDE project in the new AccuRev workspace:



Workspace creat...

Workspace created.

Workspace update

Workspace updated.

Checkout from Version Control

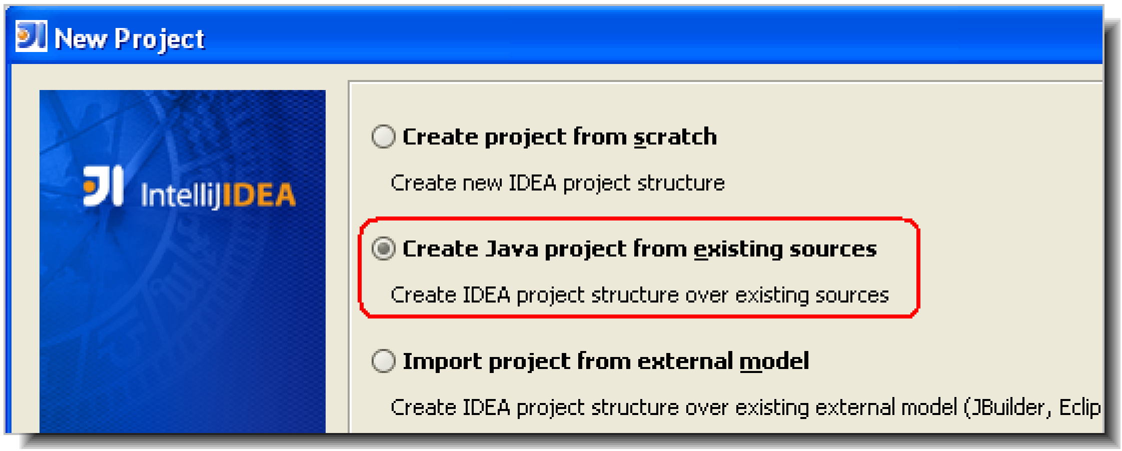
Would you like to create an IntelliJ IDEA project for the sources you have checked out to C:\wks\MoneyProject_i2?

If you are using IntelliJ IDEA 6, you must create an IDE project manually in the new AccuRev workspace. See *Creating a New IDE Project in a Workspace* on page 10. (You won't need to do Step 6, because the project's files have already been placed under version control.)

Creating a New IDE Project in a Workspace

This is essentially the standard Intellij IDEA procedure for creating a new project, with a little extra work at the end.

1. Choose **Create New Project** from the Quick Start menu, or invoke the command **File > New Project** on the IDE's main menu.
2. Choose the "from existing sources" method to create the project:

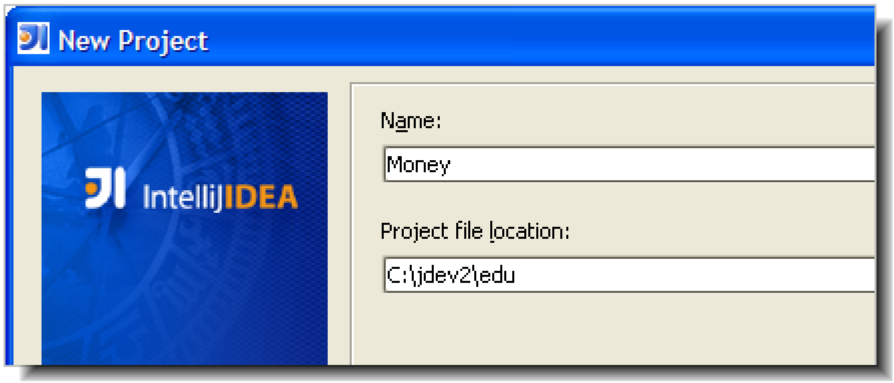


○ **Create project from scratch**
Create new IDEA project structure

● **Create Java project from existing sources**
Create IDEA project structure over existing sources

○ **Import project from external model**
Create IDEA project structure over existing external model (JBuilder, Eclip

3. Enter a project name, and specify a **Project file location** that is within an existing AccuRev workspace. You can create the project in the workspace's top-level directory, or in a subdirectory.



Name: Money

Project file location: C:\jdev2\edu

4. Continue through the wizard's screens, ending with **Finish**.

5. Make sure that AccuRev is the version control provider for the new project — see *Making AccuRev the Version Control Provider for an Existing IDE Project* on page 7.

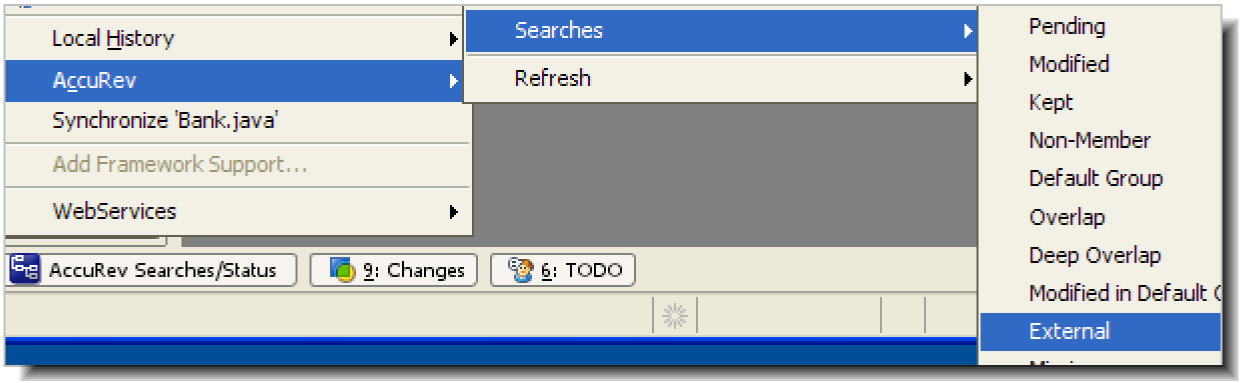
6. Make sure that the files in the project are version-controlled elements — see *Converting (external)-Status Files to Version-Controlled Elements* on page 11.

Converting (external)-Status Files to Version-Controlled Elements

AccuRev does not automatically version-control every file within an AccuRev workspace. For example, version-controlling text-editor backup files would be a waste of resources. Initially, files created in, or copied into, a workspace have **(external)** status. You can convert such files to version-controlled elements with the Integration's **Add to AccuRev Depot** command.

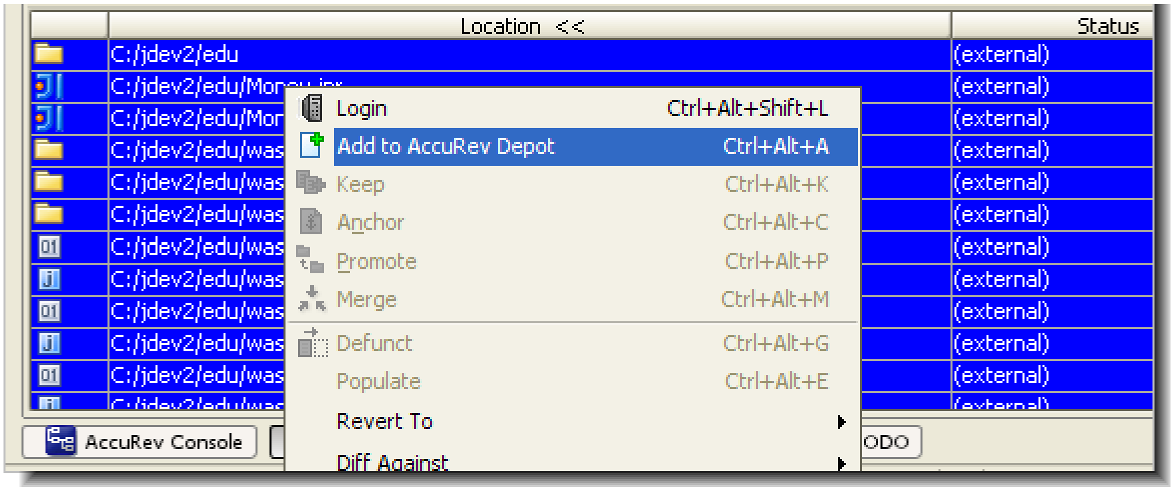
Use the following procedure when you've copied a set of files into a workspace (or, perhaps, unpacked a ZIP archive containing development data).

1. Invoke the command **AccuRev > Searches > External** to list the workspace's external files at the bottom of the IDE window.



2. Select the **(external)**-status files that you want to convert to version-controlled elements. (The **Edit > Select All** command might be useful.)

3. Invoke the command **Add to AccuRev Depot**.



The newly created elements have **(kept)** status.

Executing AccuRev Commands

You invoke AccuRev commands in the IDE from the **AccuRev** menu. This menu is available under **Version Control** on the IDE's main menu, or from context menus in several places within the IDE, including the Project, Version Control, and AccuRev Searches/Status tool subwindows.

To execute an AccuRev command:

1. Right-click a file or directory, or select multiple items, then right-click the selection.
2. Choose **AccuRev** from the context menu that appears.
3. Select an AccuRev command from the context menu, or from a cascading submenu. (Commands are enabled only if they are valid for your selection.)

The AccuRev commands are described just below, in section *AccuRev Command Reference*. Certain of the IDE's own commands (that is, commands that are not on the **AccuRev** menu) automatically invoke AccuRev commands, in order to keep the IDE synchronized with AccuRev. This is discussed in section *IDE 'Namespace' Commands and AccuRev* on page 23.

AccuRev Command Reference

The following commands are available in the IDE on the AccuRev menu, and are described in the sections below:

Login
Add to AccuRev Depot
Keep
Anchor
Promote
Merge
Defunct
Populate
Revert To
Diff Against
Synchronize Time
Update AccuRev Workspace
AccuRev History
AccuRev Statuses
Searches
Refresh

Login

If AccuRev is configured to use the *accurev_login* user-authentication scheme, you must log in before performing any AccuRev operation on a file in an AccuRev workspace. If the login is invalid or fails, then no AccuRev operation can be performed.

Add to AccuRev Depot

Converts the selected file(s) in the IDE project into AccuRev version-controlled elements. The directory containing the files is also converted to an element, if necessary.

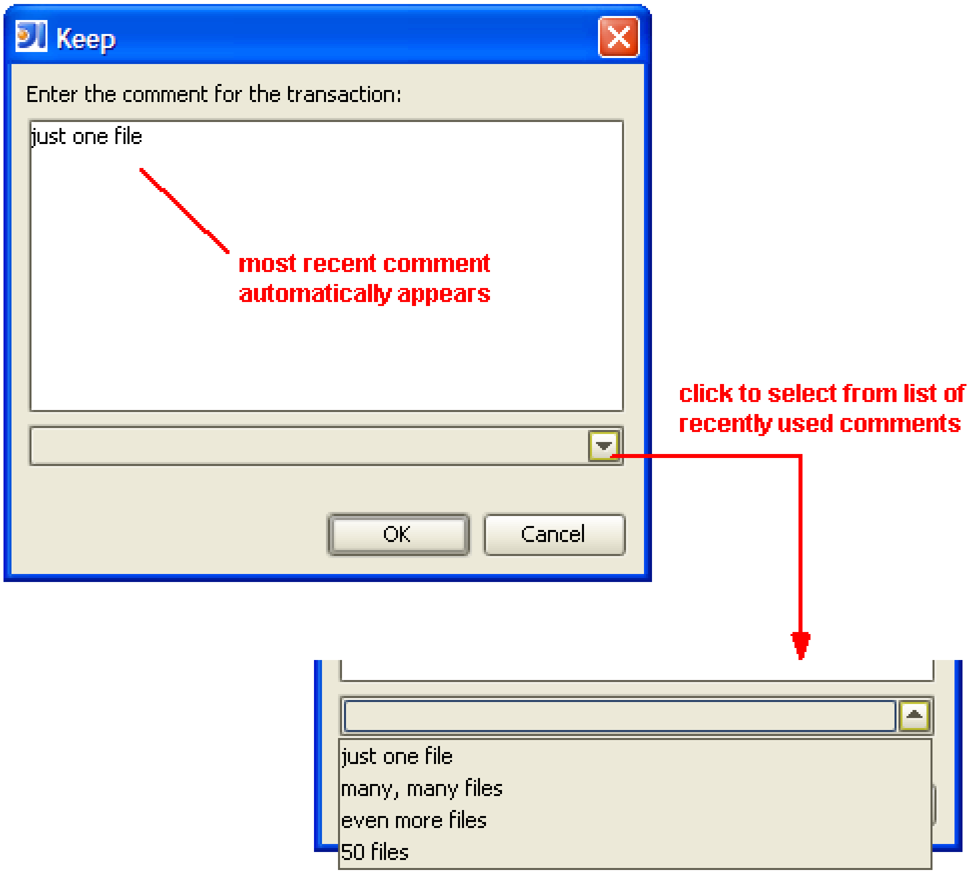
You can undo the effect of this command with the **Revert to > Backed Version** command. (See *Revert To* on page 16.) This removes the selected files from the IDE project, if you have not yet *Promote*d them.

Note: It is not necessary to place the IDE project files (**.iws**, **.ipr** and **.iml**) under version control.

Keep

Saves the changes you've made to one or more files as “private” versions in the AccuRev repository. These versions are visible only in your workspace — not in the backing stream or in other users' workspaces.

A dialog box appears, in which you can enter a comment string to be stored as part of the Keep transaction. The dialog makes it easy to reuse comment strings that you previously specified.



Enter the comment for the transaction:

just one file

most recent comment automatically appears

click to select from list of recently used comments

just one file
many, many files
even more files
50 files

Don't confuse the versions of a file created by **Keep** with the “local history” copies of the file created when you close the Editor. Local history copies are maintained by the IDE in the local file system; versions created with **Keep** exist permanently within the AccuRev repository.

Anchor

Activates the selected file(s) in your workspace, by adding them to the workspace’s default group. This ensures that the *Update AccuRev Workspace* command won’t overwrite the file with a version created in another workspace.

If the AccuRev workspace uses the AccuRev exclusive file locking or anchor-required feature, files are initially read-only. (Exclusive file locking can also be enabled for individual files.) In this situation, you *must* **Anchor** a file before editing it, in order to make it writable. These features can be enabled during workspace creation — see *Creating a New Workspace for Sources that are Already Under AccuRev Version Control* on page 8.

If you use a standard AccuRev workspace, in which files are always writable, you'll rarely need to invoke **Anchor**.

Promote

Converts one or more "private" versions into "public" versions. That is, it takes versions that you previously created in your workspace with **Keep**, and sends them to the backing stream shared by you and other members of your development team.

Implicit Keep and Promote

If you execute a **Promote** command on files which have changed since they were last preserved in the repository with **Keep**, meaning they have **(modified)** status, the Integration invokes a sequence of AccuRev commands on the selected files:

1. Performs a **Keep** on all the files in the selection with **(modified)** status.
2. Performs a **Promote** on all the files in the selection that now have **(kept)** status.

Merge

Combines two versions of an element: your version and the version in the workspace's backing stream. This command is the equivalent of "Resolve" in other source code management terminologies.

If the element type is **binary**, you can elect to **Keep** your version or the backed version. AccuRev does not support merging binary files.

If the element type is **text**, executing **Merge** brings up a merge tool. By default, the merging of text-file contents is performed by AccuRev's own Merge tool, which uses a 3-way merge algorithm.

A successful **Merge** ends with a **Keep** command, creating a new version of a file in your workspace.

> Note: AccuRev uses a separate executable to perform Merge operations. Check that your environment variables and system path are correctly defined so this executable is available to the Integration.

Defunct

Deletes a file from your disk, and marks it as having **(defunct)** status in the AccuRev workspace. A comment dialog displays, allowing you to enter comments for the defuncted element.

Other users will continue to see the file in their workspaces. When you **Promote** a defunct element to the backing stream, it disappears entirely from the workspace stream, and from the File Browser display. The element then becomes **(defunct)**, hence also active, in the backing stream.

Note: The IDE’s **Delete** command does not invoke AccuRev’s **Defunct** command. See *Using the IDE’s Delete Command* on page 24.

Populate

Restores a file with **(missing)** status to your project. To locate such files, use the Missing element search — see *Searches* on page 21.

Invoking the **Populate** command copies the version currently in the workspace stream (in the AccuRev repository) to the workspace. For more information, see “What’s the Difference between Populate and Update?” in *AccuRev Technical Notes*.

Note: don’t invoke **Populate** on a file that is not missing. This command overwrites the file, equivalently to the **Revert To > Most Recent Version** command.

Revert To

Discards the changes you've made to a file and restores another version to the workspace.

- **Revert to > Backed Version**

  For each selected element, discards all changes since the last time you promoted the element, or since the last *Update AccuRev Workspace* command, whichever is more recent.

- **Revert To > Most Recent Version**

  Discards content changes for each selected element since the last private version you created with the *Keep* command. This command does not discard changes to an element’s name.

Diff Against

Compares your file with another version. Only text files can be compared.

- **Diff Against Most Recent Version**

  Compares your file with the last private version you created with **Keep.**

- **Diff Against Backed Version**

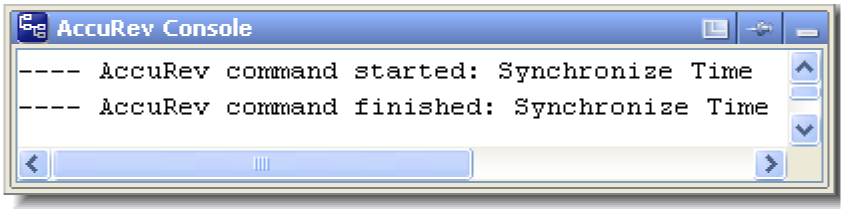
  Compares your file with the version currently in the workspace's backing stream.

- **Diff Against Basis Version**

  Compares your file with the nearest ancestor version that either was created in another workspace or was promoted from this workspace to the backing stream.

Note: AccuRev uses a separate executable to perform Diff operations. Check that your environment variables and system path are correctly defined so this executable is available to the Integration.

Synchronize Time



```
---- AccuRev command started: Synchronize Time
---- AccuRev command finished: Synchronize Time
```

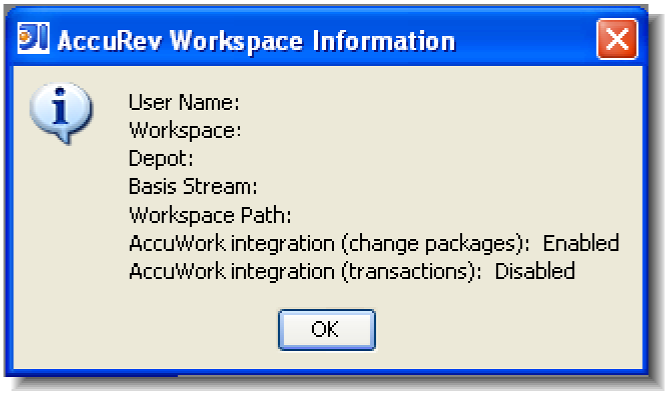
Changes the system clock on the local machine to match the clock on the AccuRev server machine. A success (or failure) message appears in the AccuRev Console window.

> Note: You must have **root** (on Linux/Unix) or **Administrator** (on Windows) privileges to change the system time.

Some AccuRev commands are disabled if the client's time doesn't match the server's time within a preset tolerance. In AccuRev Version 4.7, the set of commands that are sensitive to system clock discrepancies has been made significantly smaller.

AccuRev Workspace Information

Displays information regarding your current AccuRev work environment in a dialog box:



User Name:
Workspace:
Depot:
Basis Stream:
Workspace Path:
AccuWork integration (change packages): Enabled
AccuWork integration (transactions): Disabled

Update AccuRev Workspace

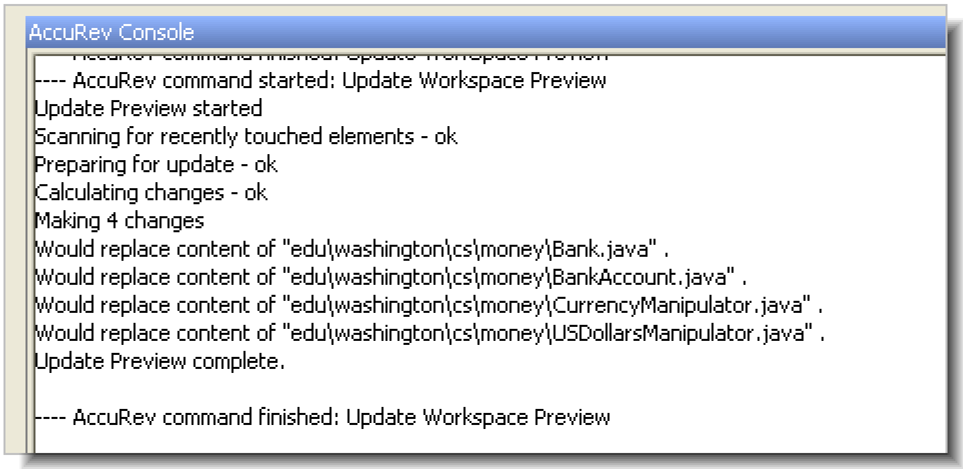
Copies versions from your workspace's backing stream into your workspace. This has the effect of incorporating other people's changes, which they have promoted to the backing stream, into your workspace.

- **Update AccuRev Workspace > Entire Workspace**

  Updates the entire workspace. This command writes "started" and "finished" messages to the AccuRev Console subwindow, but does not include element-by-element details.

- **Update AccuRev Workspace > Preview**

  Reports to the AccuRev Console subwindow the element-by-element changes that would be made to the workspace during an actual update.



AccuRev Console

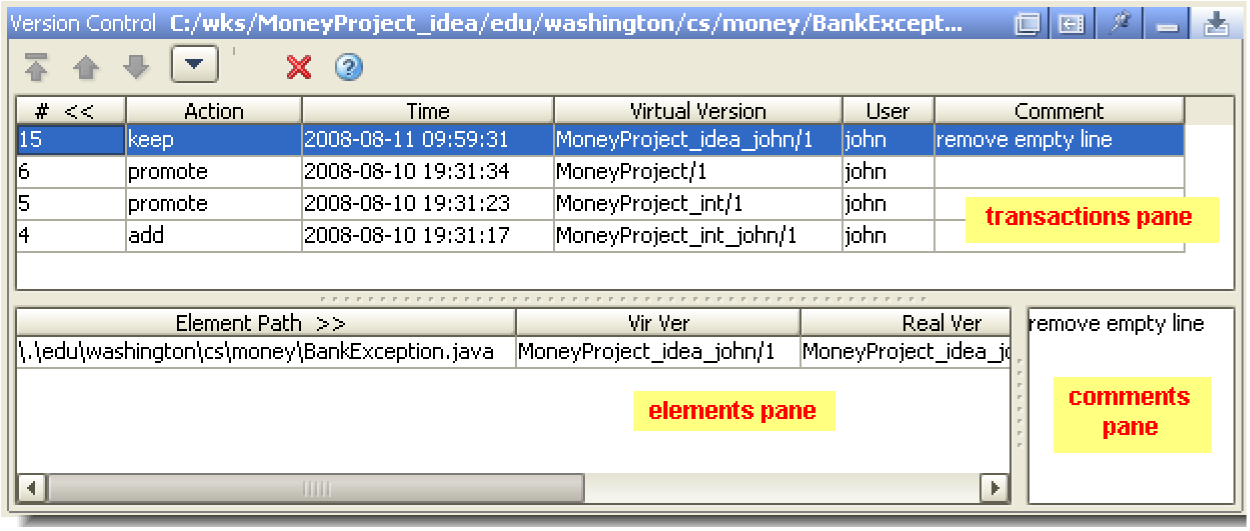
```
---- AccuRev command started: Update Workspace Preview
Update Preview started
Scanning for recently touched elements - ok
Preparing for update - ok
Calculating changes - ok
Making 4 changes
Would replace content of "edu\washington\cs\money\Bank.java" .
Would replace content of "edu\washington\cs\money\BankAccount.java" .
Would replace content of "edu\washington\cs\money\CurrencyManipulator.java" .
Would replace content of "edu\washington\cs\money\USDollarsManipulator.java" .
Update Preview complete.

---- AccuRev command finished: Update Workspace Preview
```

AccuRev History

> Note: This command applies to only one element at a time. It is disabled in the AccuRev menu if you select more than one item.

Displays the transactions for the selected file or directory element in the Version Control tool window:



Version Control C:/wks/MoneyProject_idea/edu/washington/cs/money/BankExcept...

# <<	Action	Time	Virtual Version	User	Comment
15	keep	2008-08-11 09:59:31	MoneyProject_idea_john/1	john	remove empty line
6	promote	2008-08-10 19:31:34	MoneyProject/1	john	
5	promote	2008-08-10 19:31:23	MoneyProject_int/1	john	
4	add	2008-08-10 19:31:17	MoneyProject_int_john/1	john	

Element Path >>	Vir Ver	Real Ver
\.\edu\washington\cs\money\BankException.java	MoneyProject_idea_john/1	MoneyProject_idea_jo

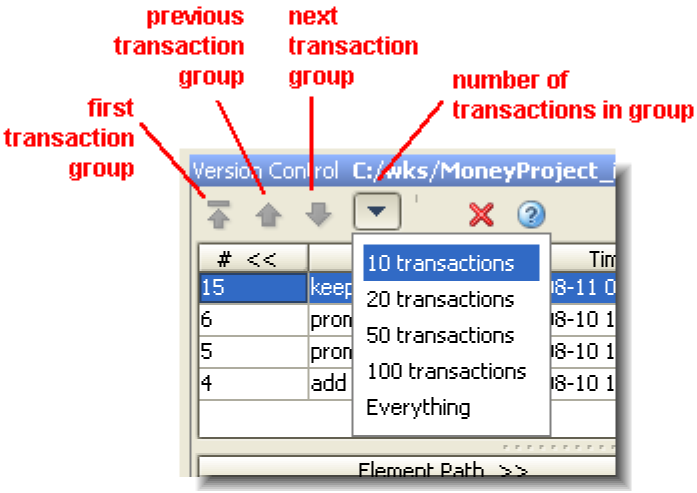
remove empty line

AccuRev tracks the complete history of each version-controlled element (file, directory, or link). This history consists of the set of transactions involving that element. Most changes to a depot are logged by transactions.

The top pane of the Version Control tool window (**transactions pane**) shows the transactions that this element was part of. If you select a transaction in the top pane, a list of all elements involved in that transaction appear in the bottom pane (**elements pane**). The comment for the selected transaction is displayed separately in a **comments pane** to the right of the Elements pane.

Typically, most of an element's transactions are created by **Keep** and **Promote** commands, as shown in the Action column in the transactions pane. Transactions are also logged in several other situations — for example:

- When an element is first added to an AccuRev depot (**add**).
- When you rename it or move it to a different directory (**move**). This action occurs when you use the IDE to move or rename a file, for example in a refactoring operation.
- When you incorporate someone else’s changes into your work (**merge**), and some others.



The toolbar at the top of the Version Control tool window contains a drop-down list that controls how many transactions are displayed at a time, and navigation buttons to move among groups of transactions: first group, previous group, and next group.

Right-clicking an item in the transactions pane or elements pane displays a context menu with the following commands. In the elements pane, the command operates on the selected version of the element. In the transactions pane, the command operates on the selected transaction’s version of the element whose history is being displayed.

- **Open**

  Opens the selected element version in the IDE Editor.

- **Export**

  Copies the selected version to a location you specify.

- **Diff Against > Other Version**

  (*Transactions pane only*) Compares the selected version to another version that you select. Click on another transaction to specify the version created by that transaction.

- **Diff Against > Previous Transaction**

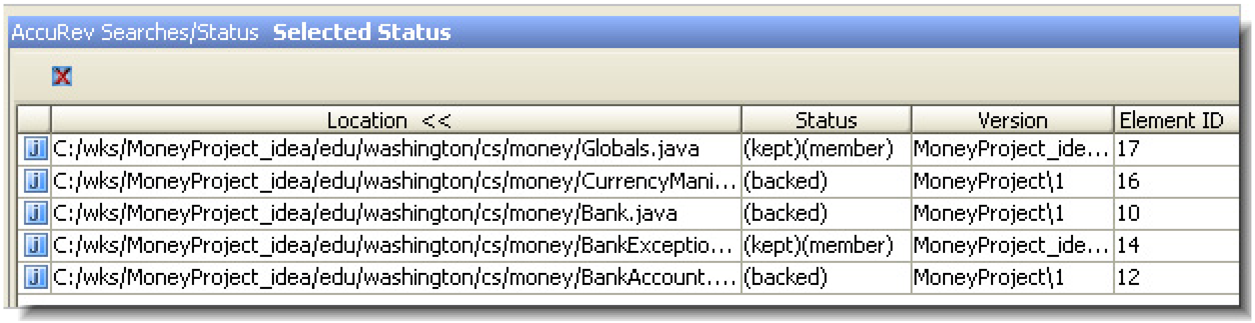
  (*Elements pane only*) Compares the selected version with the version from the previous transaction.

- **Properties**

  (*Elements pane only*) Shows the element type, element ID, and pathname within the workspace of the selected element.

AccuRev Statuses

Determines the current AccuRev status of the selected item(s), and displays the results in the AccuRev Search/Statuses subwindow.



AccuRev Searches/Status **Selected Status**

	Location <<	Status	Version	Element ID
J	C:/wks/MoneyProject_idea/edu/washington/cs/money/Globals.java	(kept)(member)	MoneyProject_ide...	17
J	C:/wks/MoneyProject_idea/edu/washington/cs/money/CurrencyMani...	(backed)	MoneyProject\1	16
J	C:/wks/MoneyProject_idea/edu/washington/cs/money/Bank.java	(backed)	MoneyProject\1	10
J	C:/wks/MoneyProject_idea/edu/washington/cs/money/BankExceptio...	(kept)(member)	MoneyProject_ide...	14
J	C:/wks/MoneyProject_idea/edu/washington/cs/money/BankAccount....	(backed)	MoneyProject\1	12

You can select one or more elements in this window and right-click to display the **AccuRev** context menu and perform any valid AccuRev command.

You can click on a column header to sort the listing on that column. Click again to reverse the sort order.

Icon Decorations in the Project Tool

Each file or directory object displayed in the Project tool subwindow has an AccuRev-level status, expressed as a set of status indicators – for example, **(kept)(member)**. (See *AccuRev Element Status* on page 26.) The Project tool does not display these status indicators, but it does display corresponding icon decorations:

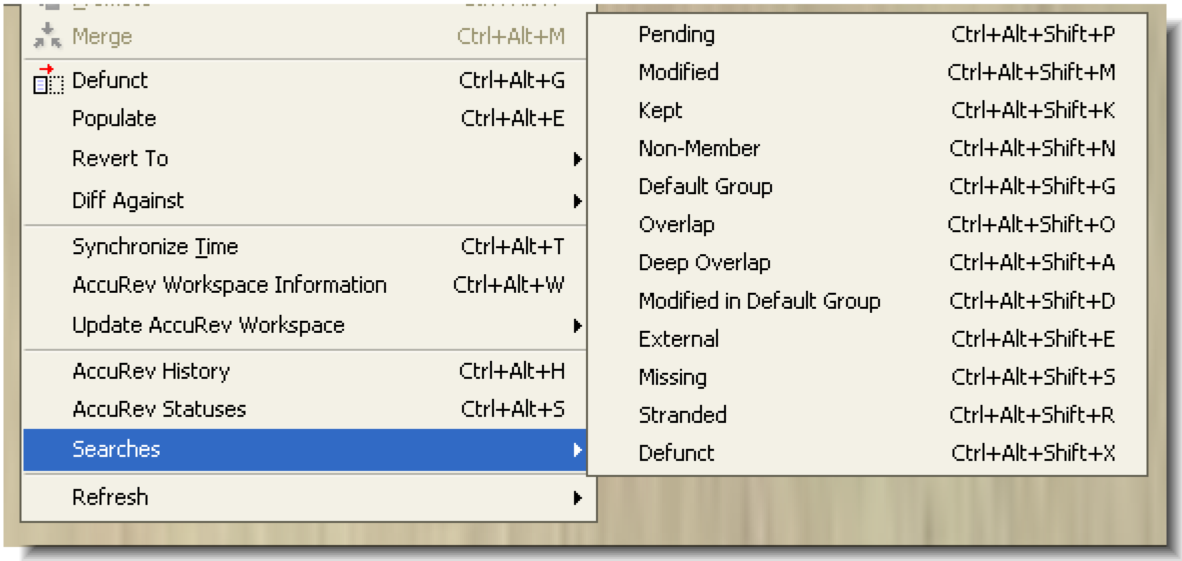
Status	Icon Decoration
(backed)	📄▲
(stale)	📄▲
(modified)	📄▼
(kept)	📄▶
(overlap)	📄◣
(member)	📄▶

The AccuRev status indicators are not mutually exclusive — for example, a file's status can include both the **(modified)** and **(member)** indicators. But the IDE uses only one decoration on each icon and/or each filename. The following precedence order of the AccuRev statuses determines which decoration appears:

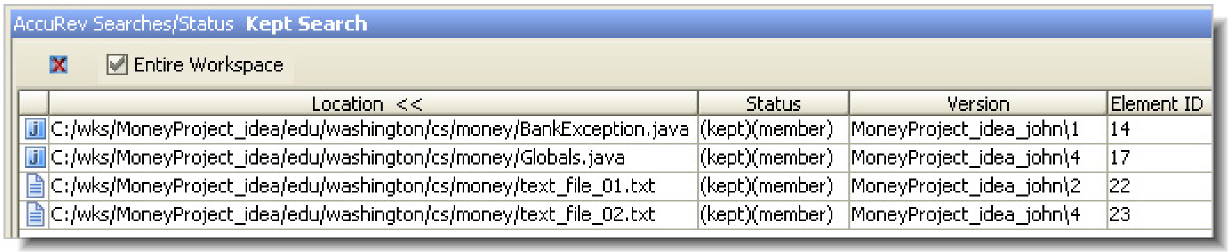
(overlap)	*highest precedence*
(stale)	
(missing)	
(modified)	
(member)	
(backed)	*lowest precedence*

Searches

Searches your workspace for files whose AccuRev status matches the criterion you specify — **Pending**, **Modified**, and so on:


Merge Ctrl+Alt+M
Defunct Ctrl+Alt+G
Populate Ctrl+Alt+E
Revert To
Diff Against
Synchronize Time Ctrl+Alt+T
AccuRev Workspace Information Ctrl+Alt+W
Update AccuRev Workspace
AccuRev History Ctrl+Alt+H
AccuRev Statuses Ctrl+Alt+S
Searches
Refresh
Pending Ctrl+Alt+Shift+P
Modified Ctrl+Alt+Shift+M
Kept Ctrl+Alt+Shift+K
Non-Member Ctrl+Alt+Shift+N
Default Group Ctrl+Alt+Shift+G
Overlap Ctrl+Alt+Shift+O
Deep Overlap Ctrl+Alt+Shift+A
Modified in Default Group Ctrl+Alt+Shift+D
External Ctrl+Alt+Shift+E
Missing Ctrl+Alt+Shift+S
Stranded Ctrl+Alt+Shift+R
Defunct Ctrl+Alt+Shift+X

The results appear in the AccuRev Searches/Status window, replacing the current contents (if any):



☑ Entire Workspace

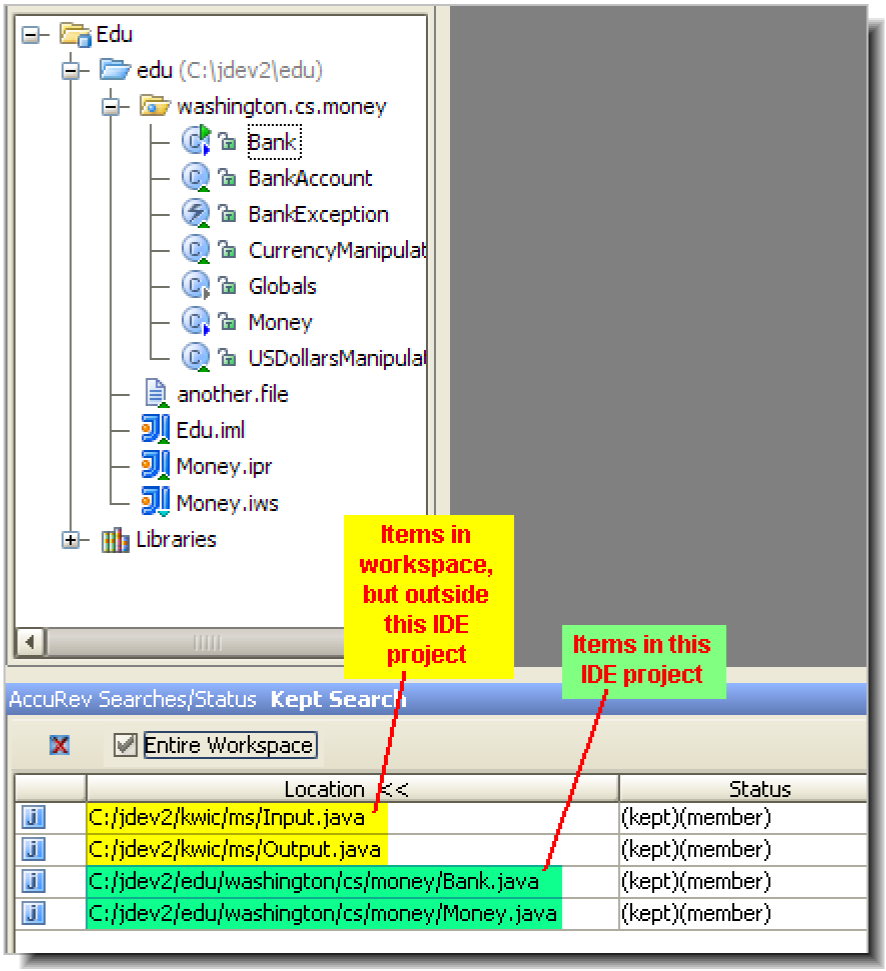
Location <<	Status	Version	Element ID
C:/wks/MoneyProject_idea/edu/washington/cs/money/BankException.java	(kept)(member)	MoneyProject_idea_john\1	14
C:/wks/MoneyProject_idea/edu/washington/cs/money/Globals.java	(kept)(member)	MoneyProject_idea_john\4	17
C:/wks/MoneyProject_idea/edu/washington/cs/money/text_file_01.txt	(kept)(member)	MoneyProject_idea_john\2	22
C:/wks/MoneyProject_idea/edu/washington/cs/money/text_file_02.txt	(kept)(member)	MoneyProject_idea_john\4	23

You can select one or more elements in this window and right-click to display the **AccuRev** context menu and perform any valid AccuRev command. This makes it easy to perform such operations as:

- Promoting elements that are pending promotion.
- Keeping and Promoting (**modified**) elements.
- Promoting elements in the workspace's default group.

- Placing (**external**) files under version control.

By default, a search is conducted through the entire workspace containing the IDE project. If the project occupies a subtree of the workspace, the search results might contain files that are not in the IDE project. Clear the **Entire Workspace** checkbox to restrict the search scope to the IDE project only.



Items in workspace, but outside this IDE project

Items in this IDE project

☑ Entire Workspace

	Location <<	Status
j	C:/jdev2/kwic/ms/Input.java	(kept)(member)
j	C:/jdev2/kwic/ms/Output.java	(kept)(member)
j	C:/jdev2/edu/washington/cs/money/Bank.java	(kept)(member)
j	C:/jdev2/edu/washington/cs/money/Money.java	(kept)(member)

Refresh

Makes a call to the AccuRev Server and refreshes the statuses of the selected files and directories. Two options are available.

- **Refresh > Selected Content Roots**

  Refreshes the selection and (if it includes a directory) the subdirectories.

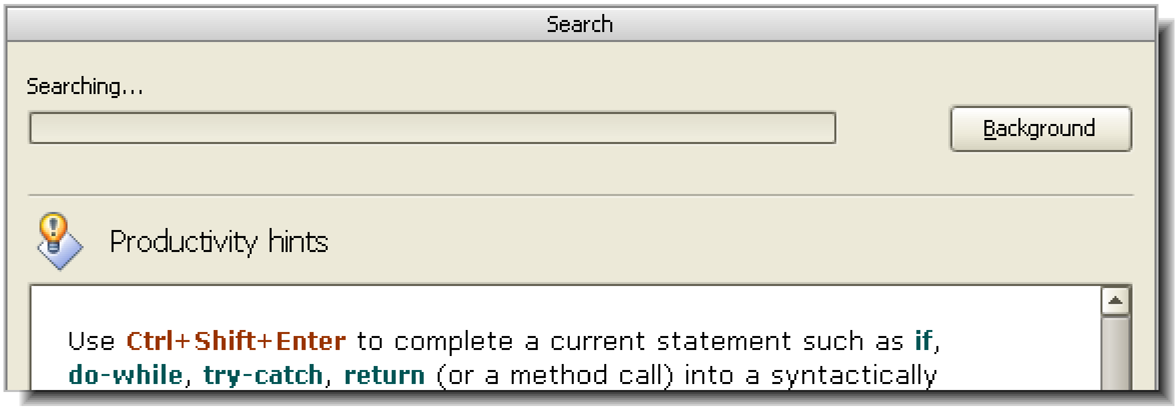
- **Refresh > Project**

  Refreshes the status of all AccuRev-controlled elements in the project.

The AccuRev icon decorations of the selected items are refreshed, too. If you have version-controlled items that are modified by the IDE or an external process, this command can help you to detect such changes to those items.

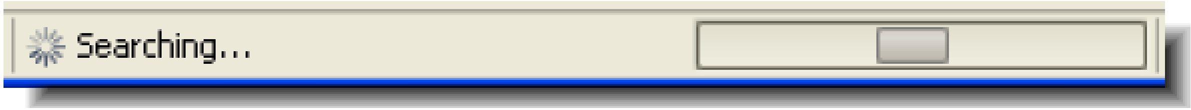
Running Commands in the Background

With IntelliJ IDEA 7 only, long-running commands — **Searches**, **Refresh**, and **Diff Against** — can be run in the background. When the command begins execution, a pop-up window appears:



Use Ctrl+Shift+Enter to complete a current statement such as if, do-while, try-catch, return (or a method call) into a syntactically

Click **Background** to close the pop-up window and displays a command-specific message and progress indicator at the bottom of the IntelliJ IDEA window:



IDE 'Namespace' Commands and AccuRev

The following sections discuss the interaction between AccuRev and the IDE commands that change a project's "namespace" — that is, commands that change rename or move a file system object, or remove the object altogether.

Renaming Version-Controlled Elements

IntelliJ IDEA provides a method for changing the pathname of a file or directory:

- To rename an object, right-click it and select **Refactor > Rename** from the object's context menu.
- To move an object to a different directory, right-click it and select **Refactor > Move** from the object's context menu.

After either of the above operations completes, the Integration automatically invokes the AccuRev **Rename** command, to record the pathname change in the repository. As usual, the results are displayed in the AccuRev Console tool window.

If you are not logged in, the IDE's **Refactor** command proceeds, but no AccuRev **Rename** command can be executed. In this case, a warning box appears, explaining that the old name will have **(missing)** status and the new name will have **(external)** status.

A similar result occurs if you use the **Move** command to move an object from one IDE project to another project that is part of the same AccuRev workspace. The object will be moved, but no

AccuRev **Rename** command will be invoked. The object will have **(missing)** status in the source location and **(external)** status in the destination location.

Using the IDE’s Delete Command

Executing the IDE’s **Edit > Delete** command on an element does *not* automatically invoke the AccuRev **Defunct** command. The element is removed from the IDE project, but gets the AccuRev status **(missing)**.

After using the **Delete** command, proceed as follows:

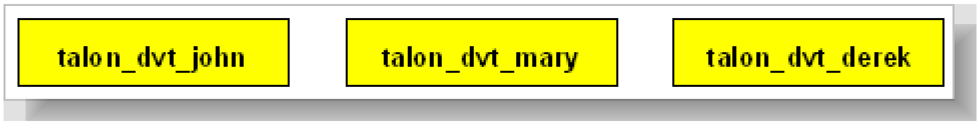
1. Invoke the command **AccuRev > Searches > Missing** to list the deleted element.
2. Invoke one of these AccuRev commands from the context menu of the deleted element:
    - The **Defunct** command removes the element at the AccuRev level, as well. (You need to Promote the element to make the removal public. You can do this after listing the element with a Defunct, Pending, or Default Group search.)
    - The **Populate** command undoes the **Delete** command, restoring the element to the IDE project. (But any recent changes that you did not preserve with **Keep** will be lost.)

Appendix: The AccuRev Usage Model

AccuRev's flexibility makes it easy to use for a variety of development scenarios. But like every software system, AccuRev has usage models that were foremost in the minds of its architects. This section describes the most common usage model.

AccuRev is a software configuration management (SCM) system, designed for use by a team of people (users) who are developing a set of files. This set of files might contain source code in any programming language, images, technical and marketing documents, audio/video tracks, etc. The files — and the directories in which the files reside — are said to be “version-controlled” or “under source control”. AccuRev also version-controls links. Version-controlled objects are termed elements.

For maximum productivity, the team's users must be able to work independently of each other — sometimes for just a few hours or days, other times for many weeks. Accordingly, each user has his own private copy of all the version-controlled elements. The private copies are stored on the user's own machine (or perhaps in the user's private area on a public machine), in a directory tree called a workspace. We can picture the independent workspaces for a three-user team as follows:



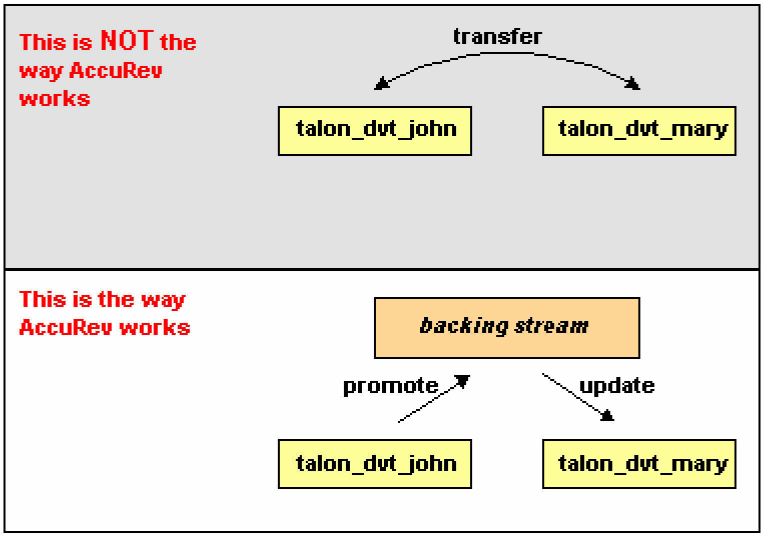
This set of users' workspaces uses the convention of having like names, suffixed with the individual usernames. AccuRev enforces this username-suffix convention. **talon_dvt** might mean “development work on the Talon product”; **john**, **mary**, and **derek** would be the users' operating system login names.

From AccuRev's point of view, development work in this set of workspaces is a continual back-and-forth between “getting in sync” and “getting out of sync”:

- Initially, the workspaces are completely synchronized: they all have copies of the same set of version-controlled elements.
- The workspaces lose synchronization as each user makes changes to some of the elements.

AccuRev uses the same synchronization paradigm as CVS — instead of transferring data directly between private areas (that is, between users' workspaces), AccuRev organizes the data transfer into two steps:

1. One user makes his changes public — available to all the other members of his team. This step is called promoting. (CVS calls this step “committing”.)
2. Whenever they wish, other team members incorporate the public changes into their own workspaces. This step is called updating.



The first step, promoting, sends your changes to a public data area, called a stream. AccuRev has several kinds of streams; the kind that we referred to is called a backing stream. The data in this public stream “is in back of” or “provides a backstop for” all the private workspaces of the team members. (The term “parent stream” is equivalent to “backing stream”.)

The second step, updating, copies other users’ changes from the backing stream to your AccuRev workspace.

AccuRev allows you to save any number of intermediate versions of an element in your workspace, before making your changes public. Such “private” versions of an element are created by the keep operation. These AccuRev-level versions are similar to, but more robust than, the set of “Local History” versions that IntelliJ IDEA maintains:

- Local History versions are temporary. (You can adjust the level of temporariness in the Preferences dialog box: **Window** > **Preferences** > **General** > **Workspace** > **Local History**).
- AccuRev versions created with keep are stored permanently in the AccuRev repository.

AccuRev Element Status

Each AccuRev element (version-controlled file, directory, or link) has a status, expressed as a set of status indicators. An element’s status addresses the question, “for this element, what is the relationship between the version in my workspace and the version in the AccuRev repository?”. There are quite a few status indicators, because AccuRev tracks an element’s status in three locations, not just two:

- The workspace tree: the directory tree on your machine, which you access as an IntelliJ IDEA project.
- The workspace stream: a corresponding area in the AccuRev repository that provides permanent storage for the private versions that you create in your workspace (for example, with the **Keep** command).
- The backing stream: the public area in the AccuRev repository that is accessed by you and other team members.

Following are descriptions of the AccuRev element status indicators.

Presence of the element in the workspace:

- **(defunct)** — the element has been marked for removal from the workspace stream with the **Defunct** command.
- **(external)** — the file or directory has not been placed under version control. (It is in the workspace tree, but not in the workspace stream.)
- **(excluded)** — the element does not appear in the workspace because it has been excluded, using the Include/Exclude facility. The AccuRev plug-in does not provide an interface to this facility.
- **(link)** — the element is a file link or a directory link.
- **(missing)** — 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 the IntelliJ IDEA **Delete** command or operating system commands.
- **(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. The AccuRev plug-in does not provide for access by element-ID.
- **(stranded)** — the element is active in the workspace, but there currently is no pathname to the element.

Changes to the element in the workspace:

- **(modified)** — the file has been modified in the workspace since the most recent **update** or **keep**.
- **(kept)** — you have created a new, private version of the element, and you have not yet made the version public with **Promote**.
- **(member)** — the element is "active" in your workspace, because you have created a new, private version of it with **Keep** or **Anchor**.

Relationship to the version in the backing stream:

- **(backed)** — an element that you’re not currently working on: the versions in the IntelliJ IDEA project (workspace tree), the workspace stream, and the backing stream are all the same.
- **(stale)** — the element needs to be updated, because the version in the backing stream has changed since the workspace’s latest **Update**.
- **(overlap)** — the element has changed both in the backing stream and in your workspace. This indicates that a **Merge** is required before you can **Promote** your changes to the backing stream.
- **(underlap)** — similar to **(overlap)**: the element has changed both in the backing stream and in your workspace, but the changes in your workspace have already been promoted to the backing stream.