AccuRev

Defining Change in SCM

The TimeSafe® Configuration Management System

# AccuRev Integrations Manual

Version 3.8

August, 2005

August 16, 2005

# AccuRev Integrations Manual
# August, 2005

# Table of Contents

# Using AccuRev with Eclipse

This note describes the integration of AccuRev with IBM's **Eclipse Platform** IDE. The integration enables users of the IDE to access AccuRev version-control facilities using the IDE's own "Team" menu.

As of AccuRev Version 3.7, this integration has been validated with: Eclipse Platform, Versions 2.1.3 and 3.0, using version 1.4.2 of the Java 2 Runtime Environment. This document describes the integration with Eclipse Platform, Version 3.0.

Check the *AccuRev Installation and Release Notes* for updated information on version and platform support. Contact AccuRev Technical Support for information on integrations with earlier versions of Eclipse Platform.

## Installing the AccuRev Plugin

The AccuRev Integration is installed in a subdirectory in the Eclipse **plugins** directory, named **com.accurev.eclipse_3.7.0**. (The final digits of the directory name may differ; we refer to it as **com.accurev.eclipse_x.x.x** below.)

Use the following procedure to install this directory in the Eclipse plugins area:

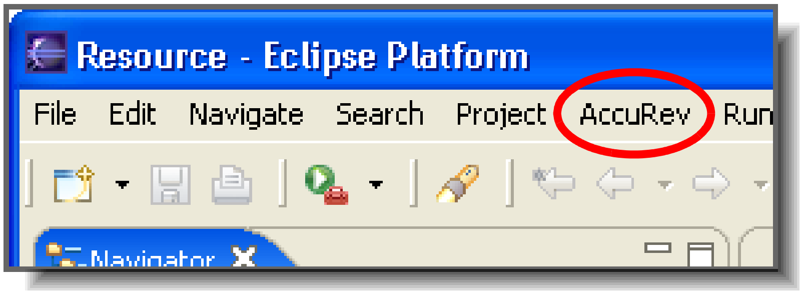
1. Verify that file **com.accurev.eclipse.zip** has been placed in the AccuRev **bin** directory during AccuRev installation.

2. Using a tool that can process ZIP files, unpack **com.accurev.eclipse.zip** to the Eclipse plugins directory. Directory **com.accurev.eclipse_x.x.x** must be a sibling of all the other Eclipse plugins. The directory hierarchy should look like this:

```
<Eclipse-installation-dir>/
   plugins/
      com.accurev.eclipse_x.x.x/
         eclipse.jar
         plugin.xml
         icons/
            accli16.gif
            anchor.gif
             ...
```

## Enabling the Integration

The integration should be enabled automatically when you start Eclipse after installing the integration JAR file. The presence of an **AccuRev** choice in Eclipse's main menu indicates that the integration is enabled.



If this menu choice does not appear, use this procedure to enable the AccuRev Integration with Eclipse:

1. Select **Window > Customize Perspective** from the Eclipse main menu.
2. Go to the **Commands** tab.
3. In the **Available command groups** column, check the **AccuRev Plugin** checkbox.
4. Click **OK**.

## Disabling the Integration

If you wish to disable use of the AccuRev Integration with Eclipse:

1. Select **Window > Customize Perspective** from the Eclipse main menu.
2. Go to the **Commands** tab.
3. In the **Available command groups** column, clear the **AccuRev Plugin** checkbox.
4. Click **OK**.

The Integration is disabled immediately, indicated by the removal of the item **AccuRev** from the Eclipse Platform's main menu.

## Troubleshooting: Interaction with the MyEclipse Plugin

MyEclipse Enterprise Workbench is a popular plugin for Eclipse. If the MyEclipse plugin is already installed, installing the AccuRev integration plugin might seem to have no effect: the Eclipse main menu doesn't include an **AccuRev** choice, and the **Commands** tab of the **Customize Perspective** dialog box doesn't include an **AccuRev Plugin** checkbox.

To avoid this problem, install the AccuRev plugin *before* the MyEclipse plugin. If you've already installed the MyEclipse plugin, use either of these workarounds:

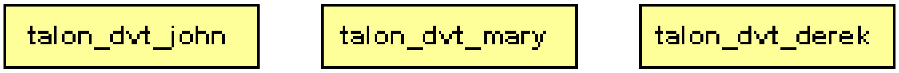
- Workaround #1: Install the AccuRev plugin, as described above. When starting Eclipse, use the **–clean** option.
- Workaround #2: Uninstall both the AccuRev and MyEclipse plugins from the Eclipse **plugins** directory. Then reinstall them both, making sure to do AccuRev first, then MyEclipse.

# 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".

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 files. 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:



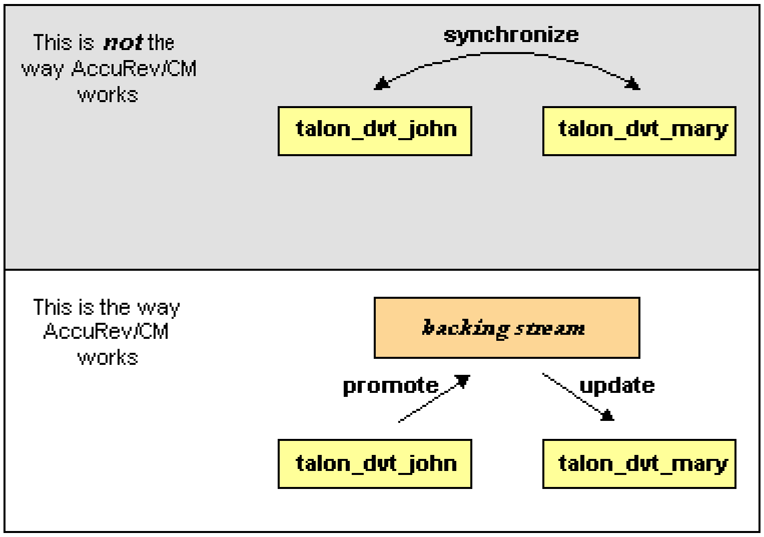
> Note: an AccuRev workspace corresponds to an Eclipse project. Be careful to distinguish between AccuRev workspaces and the overall Eclipse workspace. The AccuRev Integration is compatible with the use of multiple Eclipse workspaces.

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 perspective, 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 files.
- The workspaces lose synchronization as each user makes changes to some of the files.
- Periodically, users share their changes with each other. When **john** incorporates some or all of **mary**'s changes into his workspace, their two workspaces become more closely (perhaps completely) synchronized.

You might assume that the workspace synchronization process involves the direct transfer of data from one workspace to another. But this is not the way AccuRev organizes the work environment. 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 promotion.
2. Whenever they wish, other team members incorporate the public changes into their own workspaces. This step is called updating.

The first step involves a public data area, called a stream. AccuRev has several kinds of streams; the kind that we're discussing here 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.

AccuRev also allows you to save any number of intermediate versions of a file in your workspace, before making your changes public. Such "private" versions of a file are created by the keep operation.

## Establishing Your Identity

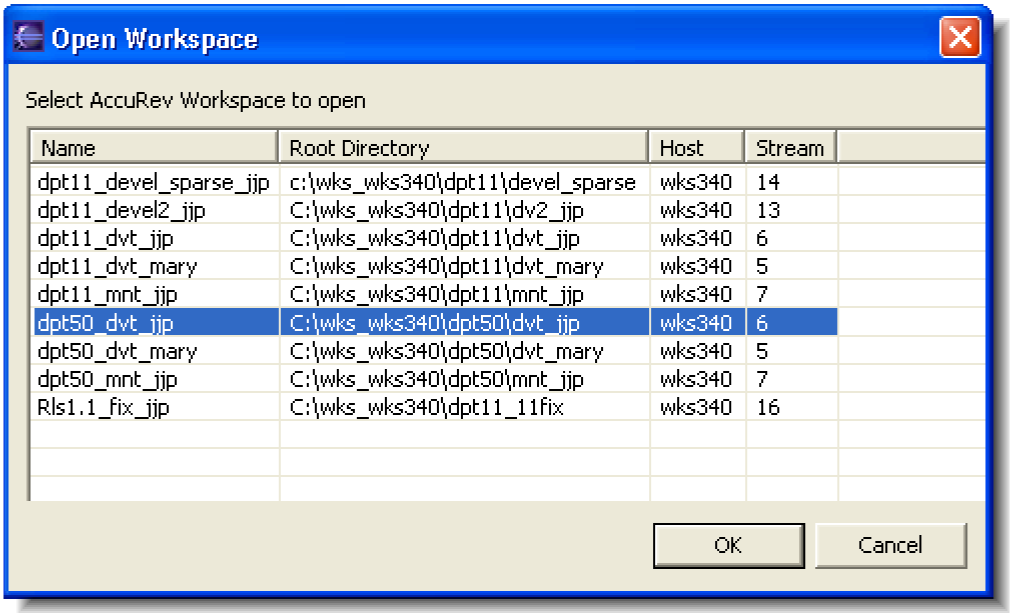
All AccuRev commands must be executed by a user who is listed in the AccuRev user registry. By default, the Integration uses your operating-system username as your AccuRev username. To have the Integration use a different AccuRev username, set environment variable ACCUREV_PRINCIPAL before starting the IDE. Here's a Bash Shell example:

```
export ACCUREV_PRINCIPAL=derekp
```

## Working with Your Data

The various Integration commands move data between the central source-code repository (called a depot) and your personal work area (called a workspace in AccuRev, and a project in Eclipse). The AccuRev workspace must be created with the AccuRev GUI or CLI; there is no Integration command to create an AccuRev workspace.

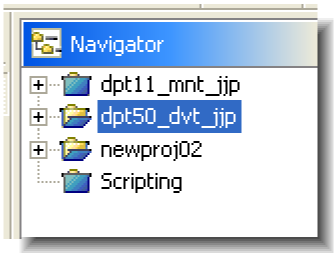
To work with the files in a particular AccuRev workspace, you must load it into your Eclipse workspace. This is the equivalent of opening an Eclipse project with the command **Project > Open Project**. The command **AccuRev > Open AccuRev Workspace** opens a window in which you can select a workspace:



Select AccuRev Workspace to open

| Name | Root Directory | Host | Stream |
|---|---|---|---|
| dpt11_devel_sparse_jjp | c:\wks_wks340\dpt11\devel_sparse | wks340 | 14 |
| dpt11_devel2_jjp | C:\wks_wks340\dpt11\dv2_jjp | wks340 | 13 |
| dpt11_dvt_jjp | C:\wks_wks340\dpt11\dvt_jjp | wks340 | 6 |
| dpt11_dvt_mary | C:\wks_wks340\dpt11\dvt_mary | wks340 | 5 |
| dpt11_mnt_jjp | C:\wks_wks340\dpt11\mnt_jjp | wks340 | 7 |
| dpt50_dvt_jjp | C:\wks_wks340\dpt50\dvt_jjp | wks340 | 6 |
| dpt50_dvt_mary | C:\wks_wks340\dpt50\dvt_mary | wks340 | 5 |
| dpt50_mnt_jjp | C:\wks_wks340\dpt50\mnt_jjp | wks340 | 7 |
| Rls1.1_fix_jjp | C:\wks_wks340\dpt11_11fix | wks340 | 16 |

The AccuRev workspace appears in the Navigator pane, along with any other projects you've opened.



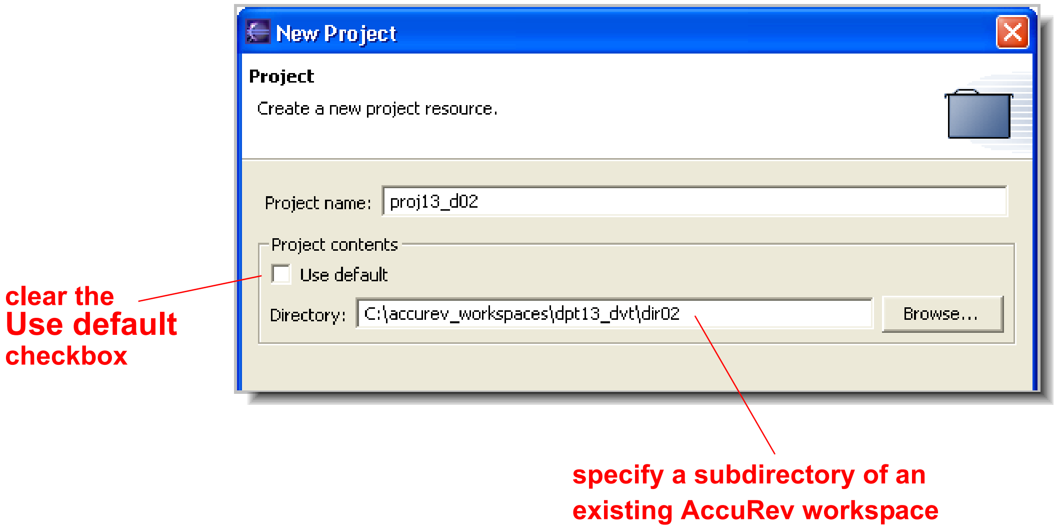
Note: opening an AccuRev workspace does not copy any data into the Eclipse **workspace** folder. The data remains in its original location, but simply becomes accessible as a project in the Eclipse workspace. In this sense, an open AccuRev workspace is like an Eclipse linked folder.

## Loading Part of an AccuRev Workspace as an Eclipse Project

You can also turn a *subtree* of an AccuRev workspace into an Eclipse project:

1. Create a new Eclipse project (**File > New > Project**), and select the **Simple > Project** wizard.

2. Fill in an appropriate project name. Instead of accepting the default location for "Project contents", specify a subdirectory of an existing AccuRev workspace.



New Project

**Project**

Create a new project resource.

Project name: proj13_d02

Project contents

□ Use default

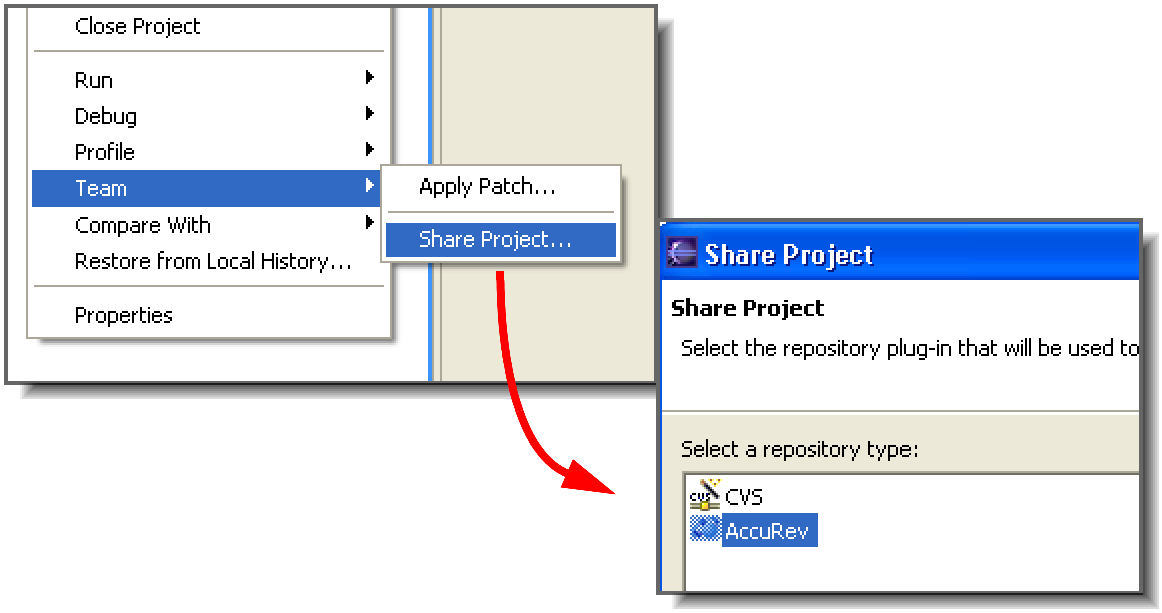
Directory: C:\accurev_workspaces\dpt13_dvt\dir02

clear the **Use default** checkbox

specify a subdirectory of an existing AccuRev workspace

3. Click **Finish** to end the wizard.

4. In the Navigator pane, right-click the new project, and select the **Team> Share Project** command from the context menu. Then select **AccuRev** as the provider.



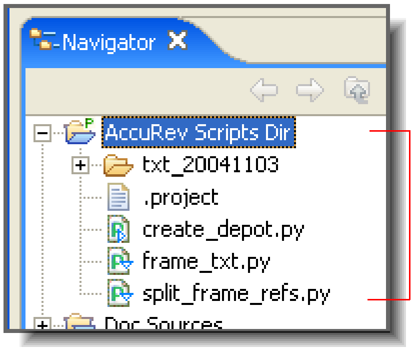
Share Project

**Share Project**

Select the repository plug-in that will be used to

Select a repository type:

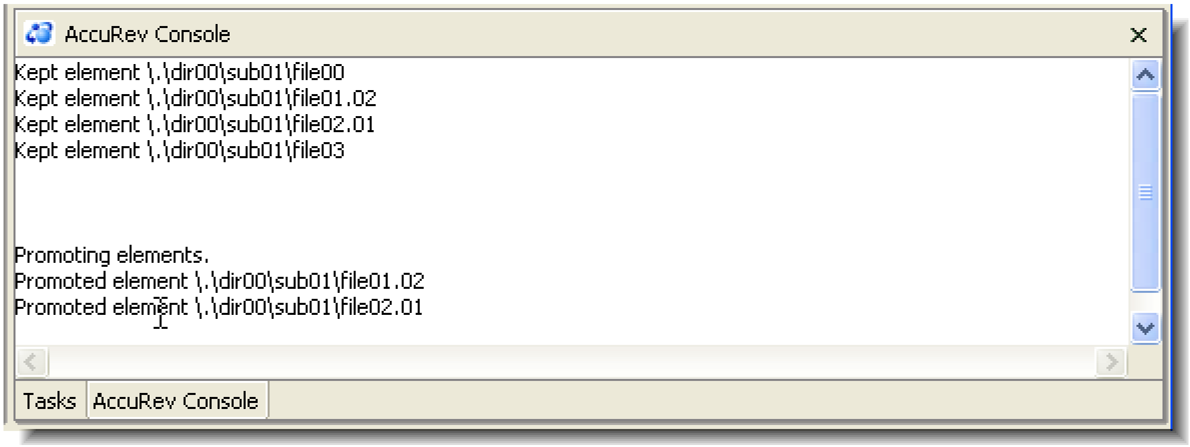
The contents of the AccuRev workspace’s subdirectory become the contents of the Eclipse project. AccuRev commands appear on the **Team** menu for the files in this project.



Eclipse project contains a subdirectory of an existing AccuRev workspace

## AccuRev Command Output

AccuRev commands generate confirmation messages, which are displayed in the **AccuRev Console** pane. This pane appears in the lower right corner of the Eclipse window; it might share the space with other items, such as the **Tasks** pane.



```
Kept element \.\dir00\sub01\file00
Kept element \.\dir00\sub01\file01.02
Kept element \.\dir00\sub01\file02.01
Kept element \.\dir00\sub01\file03



Promoting elements.
Promoted element \.\dir00\sub01\file01.02
Promoted element \.\dir00\sub01\file02.01
```

The AccuRev Console automatically appears whenever an AccuRev command produces a user message. You can also make the console appear with the command **Window > Show View > Other**: select **AccuRev > AccuRev Console** from the window that appears.

# Command Summary

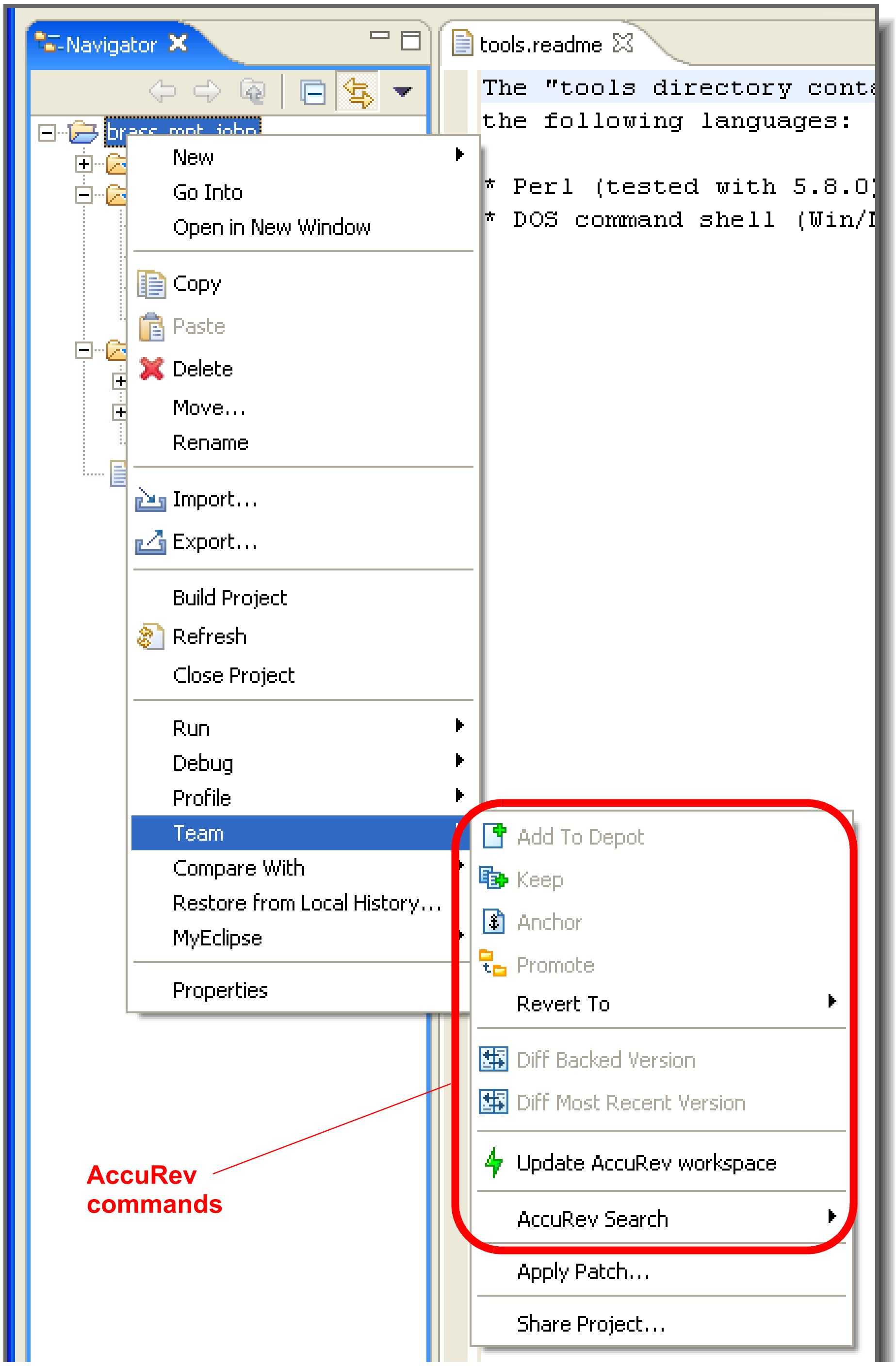
You invoke AccuRev commands through context menus:

1. Right-click a file or directory in the Navigator pane.
2. Go to the **Team** submenu.
3. Select an AccuRev command.

As this example shows, you can also select multiple files before invoking the context menu.

The AccuRev commands supported by the Integration are described in the sections below.

In addition to offering these explicit AccuRev commands, the Integration automatically invokes the AccuRev **Rename** command when you rename a version-controlled element or relocate it to a different directory. See *Changing an Element's Name* on page 12.



```
The "tools directory conta
the following languages:

* Perl (tested with 5.8.0)
* DOS command shell (Win/I
```

AccuRev commands

## Add to Depot

The **Add to Depot** command converts one or more of the files in the development project into AccuRev version-controlled elements. (The directory containing the files is also converted to an element, if necessary.) The files to be converted must be located in a directory within an existing AccuRev workspace.

Note: the Eclipse **.project** file in the workspace's top-level directory will not be placed under version control, even if you include it in a set of files submitted to this command.

## Keep

The **Keep** command 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 "public" backing stream or in other users' workspaces.

Don't confuse the versions of a file created by **Keep** with the "local history" copies of the file created when you invoke **File > Save** in an Editor pane. Local history copies exist only within the Eclipse IDE itself; versions created with **Keep** exist permanently within the AccuRev repository.

## Anchor

Many version control systems have a "check out" command that makes a file writable, so that you can edit it. But with AccuRev, your files are *always* writable. AccuRev *does* have its own similar command, called **Anchor**, This doesn't change the writability of a file, but it does activate the file in your workspace (that is, adds it to the workspace's default group). One effect of this is to ensure that the file won't be overwritten by an **Update AccuRev Workspace** command. Under normal circumstances, you rarely need to invoke the **Anchor** command.

## Promote

The **Promote** command 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.

## Revert to Backed

The **Revert to Backed** command discards the changes you've made to a file. It restores the version that was in the backing stream at the time of your most recent **Update**. (But if you promoted one or more versions of the element to the backing stream since your most recent **Update**, it restores the most recently promoted version.)

## Revert to Recent

The **Revert to Recent** command is similar to **Revert to Backed**, but rolls back a file only as far as the private version you recently created with **Keep**. This command is useful when you edit a file, save it with **File > Save**, then decide to throw away the changes.

## Diff Backed Version

The **Diff Backed Version** command compares your file with the version currently in the workspace's backing stream.

Note: only text files can be compared.

## Diff Most Recent Version

The **Diff Most Recent Version** command compares your file with the private version you recently created with **Keep**. (“What have I changed since my last **Keep**?”)

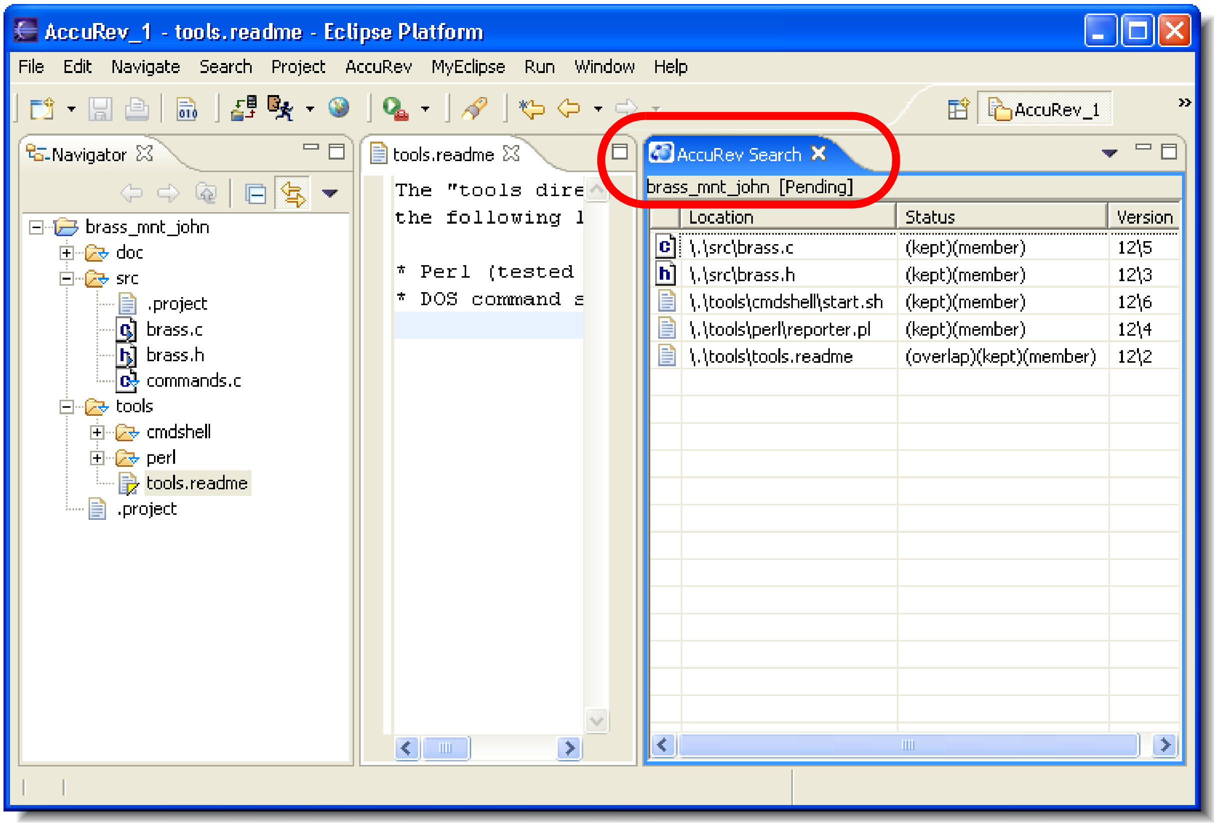
Note: only text files can be compared.

## Update AccuRev Workspace

The **Update AccuRev Workspace** command 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.

## AccuRev Search

Displays the results of a workspace search in a new pane, **AccuRev Search**. (If such a pane is already open, the new search replaces its contents.)

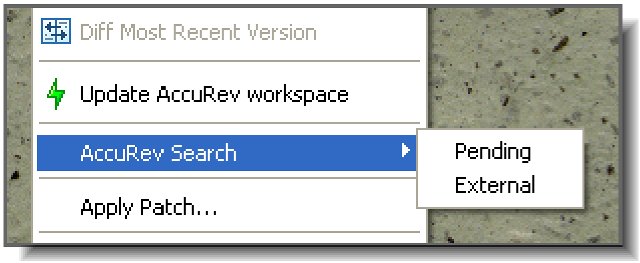


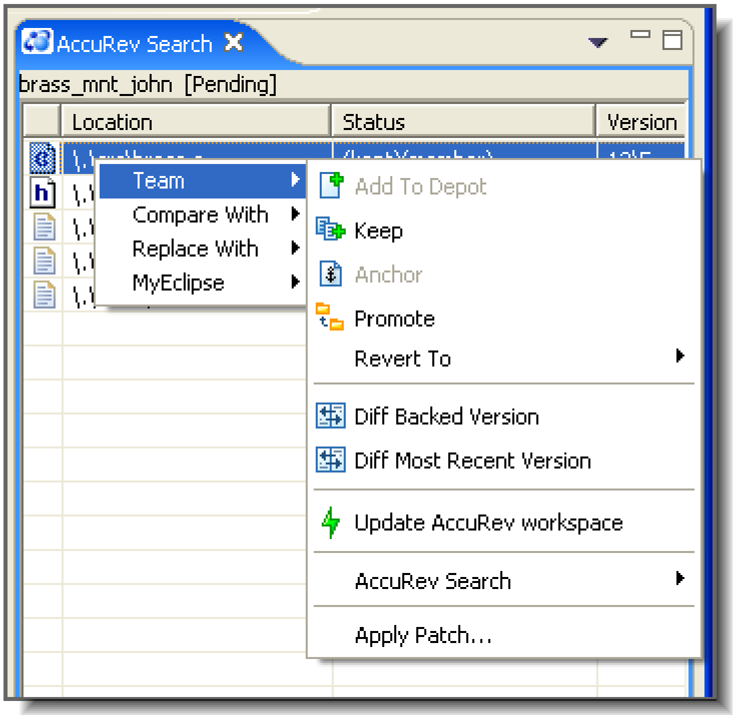
```
The "tools dire
the following l

* Perl (tested
* DOS command s
```

brass_mnt_john [Pending]

| | Location | Status | Version |
|---|---|---|---|
| c | \.\src\brass.c | (kept)(member) | 12\5 |
| h | \.\src\brass.h | (kept)(member) | 12\3 |
| | \.\tools\cmdshell\start.sh | (kept)(member) | 12\6 |
| | \.\tools\perl\reporter.pl | (kept)(member) | 12\4 |
| | \.\tools\tools.readme | (overlap)(kept)(member) | 12\2 |

This pane shows all the elements in the workspace, regardless of their directory location, that meet a particular search criterion. The AccuRev Integration with Eclipse implements two search criteria: **Pending** and **External**. (The AccuRev GUI implements additional search criteria. For a full discussion of searches see *Searches and File Statuses* on page 53 of the *AccuRev User’s Guide (GUI)*.)





You can invoke AccuRev commands on the items displayed in the AccuRev Search pane, using their context menu (**Team** submenu). This makes it easy to perform such operations as:

- ...promoting some or all of the elements that are pending promotion
- ... placing under version control some or all of the workspace's external files

Note: it is not necessary to place the Eclipse **.project** file, located in the workspace's top-level directory, under version control.

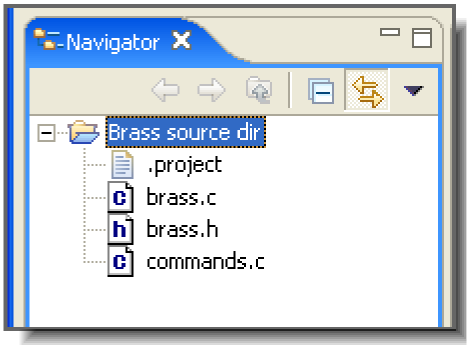
As with the AccuRev Console pane, you can use **Window** > **Show View** > **Other** to make the AccuRev Search pane appear. (This does not cause any search to be performed; the pane comes up empty.)

## Searches on Parts of a Workspace

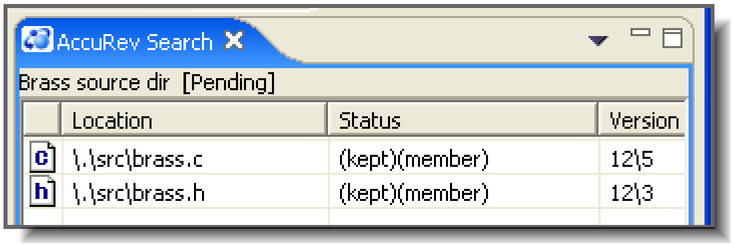
As described in *Loading Part of an AccuRev Workspace as an Eclipse Project* on page 5, you can create an Eclipse project that contains a particular subtree of an existing AccuRev workspace. For such a project, the AccuRev Search window can display either full or restricted results of a workspace search:

- **full**: displays objects throughout the entire AccuRev workspace that meet the search criterion, including objects that are not in the subtree loaded into the Eclipse project.
- **restricted**: displays only objects that both meet the search criterion and are loaded into the Eclipse project.

Use the AccuRev Search pane's **Show all elements** setting, controlled from its toolbar, to switch between the full and restricted displays. For example, suppose you've loaded just a workspace's **src** directory — not the **doc** or **tools** directory — into an Eclipse project.



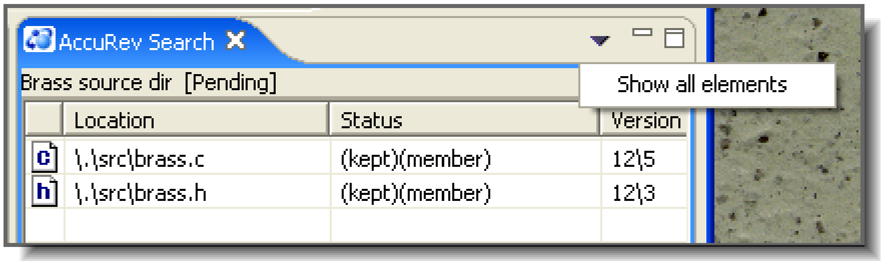
Invoking the command **Team > Show AccuRev search > Pending** displays the elements in the Eclipse project (that is, in the **src** directory) that are pending promotion. Then, turning on the **Show all elements** setting in the toolbar expands the display to include pending-promotion elements in the workspace that are not loaded into the Eclipse project.



Brass source dir [Pending]

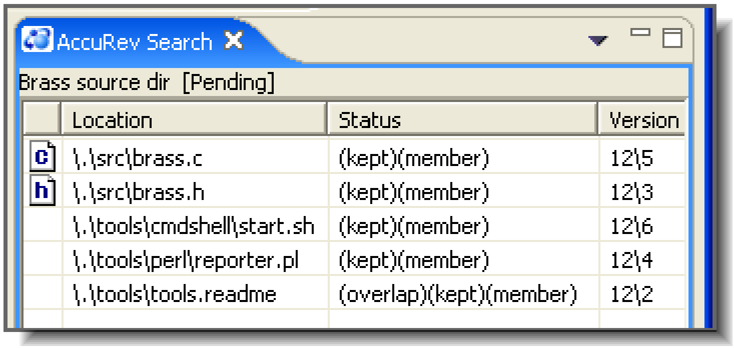
| | Location | Status | Version |
|---|---|---|---|
| c | \.\src\brass.c | (kept)(member) | 12\5 |
| h | \.\src\brass.h | (kept)(member) | 12\3 |

elements in Eclipse project (src directory) that satisfy search criterion



Brass source dir [Pending]

| | Location | Status | Version |
|---|---|---|---|
| c | \.\src\brass.c | (kept)(member) | 12\5 |
| h | \.\src\brass.h | (kept)(member) | 12\3 |



Brass source dir [Pending]

| | Location | Status | Version |
|---|---|---|---|
| c | \.\src\brass.c | (kept)(member) | 12\5 |
| h | \.\src\brass.h | (kept)(member) | 12\3 |
| | \.\tools\cmdshell\start.sh | (kept)(member) | 12\6 |
| | \.\tools\perl\reporter.pl | (kept)(member) | 12\4 |
| | \.\tools\tools.readme | (overlap)(kept)(member) | 12\2 |

elements in entire AccuRev workspace that satisfy search criterion

## Changing an Element's Name

Eclipse provides a number of methods for changing the pathname of a file or directory:

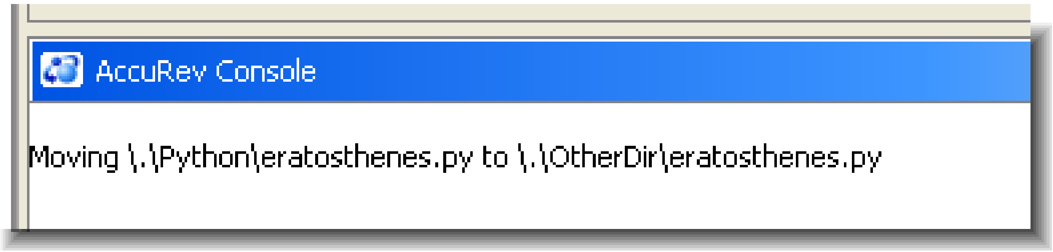
- Renaming a object:
  - Right-click the object, and select **Rename** from the context menu.

- Select the object, and press function key **F2**.

- Moving an object to a different directory:
  - Right-click the object, and select **Move** from the context menu.
  - Drag-and-drop the object from one directory to another.

  Eclipse enforces the restriction that you cannot move an object to a different Eclipse project (AccuRev workspace). This corresponds to AccuRev's own restriction in this area.

When you complete any of the above operations, the Integration automatically invokes the AccuRev Rename command, to record the change of pathname in the repository. As usual, the results are displayed in the AccuRev Console pane:



```
Moving \.\Python\eratosthenes.py to \.\OtherDir\eratosthenes.py
```

# AccuRev SCC Integration

This chapter describes the integration of AccuRev with Windows IDEs that support the Microsoft SCC (Source Code Control) programming interface. The SCC interface enables users of the tool to access AccuRev version-control facilities using the tool's own menu structure.

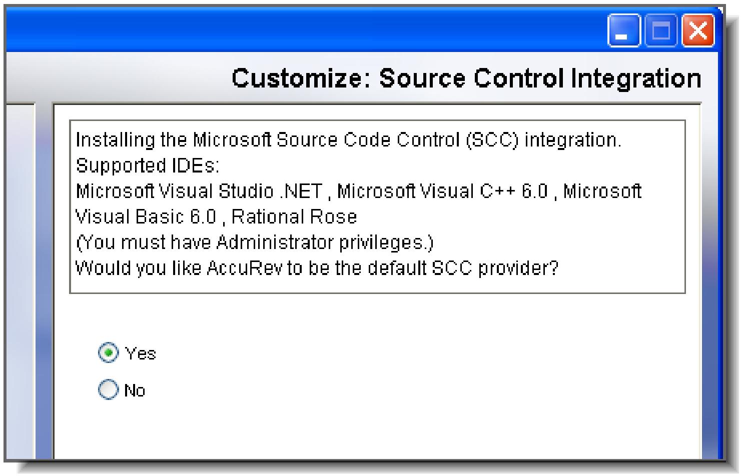
As of AccuRev Version 3.5, this integration has been designed for, and validated with, the following tools:

- Microsoft Visual Basic 6.0
- Microsoft Visual C++ 6.0
- Microsoft Visual Studio .NET
- Rational Rose

Check the AccuRev Release Notes for updated information on which IDEs the integration supports.

## Installing the Integration

Installation of AccuRev on a Windows machine automatically includes the AccuRev-SCC integration. You need only choose whether AccuRev will be the default provider of SCC services:



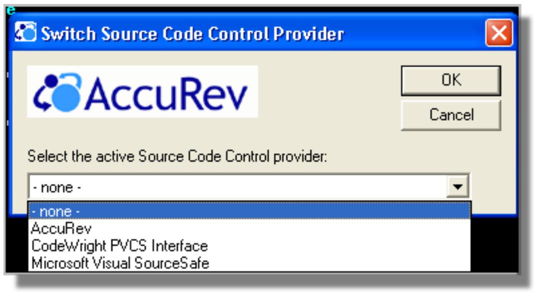
Customize: Source Control Integration

Installing the Microsoft Source Code Control (SCC) integration.
Supported IDEs:
Microsoft Visual Studio .NET , Microsoft Visual C++ 6.0 , Microsoft Visual Basic 6.0 , Rational Rose
(You must have Administrator privileges.)
Would you like AccuRev to be the default SCC provider?

● Yes
○ No

Before running the AccuRev installation program, make sure to exit all the tool(s) on your machine that use the SCC interface.

## Turning Off the Integration


Switch Source Code Control Provider
AccuRev
OK
Cancel
- none -
AccuRev
CodeWright PVCS Interface
Microsoft Visual SourceSafe

Select the active Source Code Control provider: - none -

There is no way to "uninstall" the AccuRev-SCC integration, except to uninstall AccuRev itself. But you can turn off the integration using the utility program **SwitchSCC.exe**, which is located in the AccuRev **bin** directory.

This program displays a simple dialog box. You can disable the SCC programming interface completely (select **-none-**), or you can enable another SCC provider. Only one SCC provider can be active at a time, so selecting another provider also turns off the AccuRev-SCC integration.

## Using a Workspace

The various SCC commands move data between the central source-code repository and your personal work area. Your work area must be located in an existing AccuRev workspace. You must create this workspace with the AccuRev GUI or CLI; there is no SCC command to create a workspace.

## Establishing Your Identity

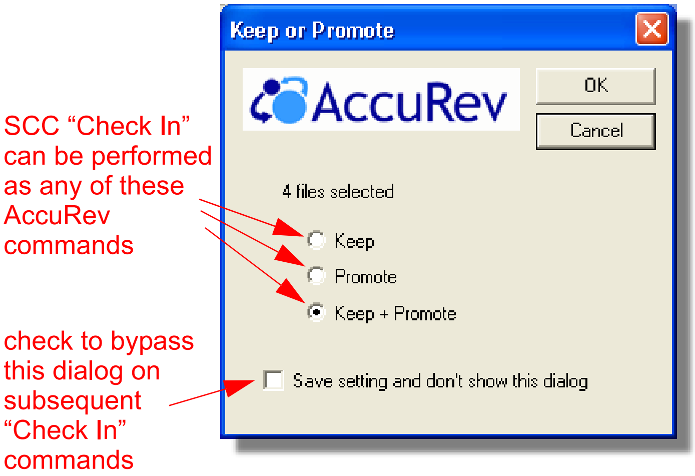
Most AccuRev commands, invoked through the AccuRev-SCC integration, must be performed by an official user. That is, you must have an AccuRev username (or "principal-name").

- Visual Basic prompts you to specify your username at the beginning of a session. To change usernames, you must end the Visual Basic session and start a new session.
- Visual C++ determines your username by examining a Windows Registry setting. To change this setting, invoke the **Tools** > **Options** command, go to the **Source Control** tab, and change the **Login** field.
- Visual Studio .NET determines your username by examining a Windows Registry setting. To change this setting, invoke the **Tools** > **Options** command, go to the **Source Control / SCC Provider** folder, and change the **Login ID** value.

## SCC Commands and AccuRev Commands

The SCC programming interface enables a predetermined set of source-code-control commands: **Add to Source Control**, **Check Out**, **Check In**, etc. When you invoke an SCC command (say, **Check In**), control is handed off to the corresponding AccuRev command.

The SCC model of source-code-control functionality does not match the AccuRev model exactly. For example, SCC's **Check In** command does not correspond exactly to a single AccuRev command. Accordingly, when you invoke **Check In**, AccuRev must determine exactly how to perform this operation. You can make a selection each time from a dialog box displayed by the integration.



4 files selected

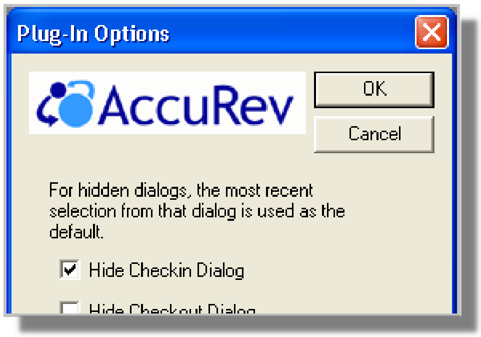
○ Keep
○ Promote
● Keep + Promote

□ Save setting and don't show this dialog

SCC "Check In" can be performed as any of these AccuRev commands

check to bypass this dialog on subsequent "Check In" commands

But for convenience, you'll probably want to configure the integration to remember how the SCC command is to be executed by AccuRev. The AccuRev-specific dialog box won't appear on subsequent invocations of the SCC command. You can configure the integration either by selecting a checkbox in the dialog box for a specific command (shown above), or by using the overall AccuRev Plug-In Options dialog (shown below).



For hidden dialogs, the most recent selection from that dialog is used as the default.

☑ Hide Checkin Dialog
□ Hide Checkout Dialog

For more on the Plug-In Options dialog, see AccuRev Plug-In Options on page 23.

## AccuRev Command Output

AccuRev commands generate confirmation messages, which are displayed within the IDE:

- In Visual Basic, the messages are displayed in a separate AccuRev Messages window. (Bugs in Visual Basic's implementation of the Source Control Results Window prevent use of that window.)
- In Visual C++, the messages are displayed in the Source Control tab of the Output window.
- In Visual Studio .NET, the SCC integration displays messages in the Source Control tab of the Output window, and AccuRev displays its own messages in a separate AccuRev Messages window.

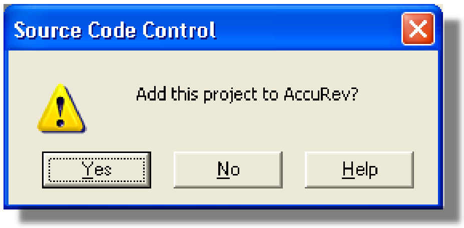
## SCC Command Summary

The sections below describe each of the SCC commands, differences (if any) between the SCC and AccuRev models, and the AccuRev-specific dialog boxes that map the SCC commands to AccuRev commands.

> Note: the **Open From Source Control** and **Add Project From Source Control** SCC commands, displayed on the **File > Source Control** menu in Visual Studio .NET, are not supported.

### Add to Source Control

The SCC **Add to Source Control** command converts one or more of the files in the development project into AccuRev version-controlled elements. (The directory containing the files is also converted to an element.) The files to be converted must be located in a directory within an existing AccuRev workspace.

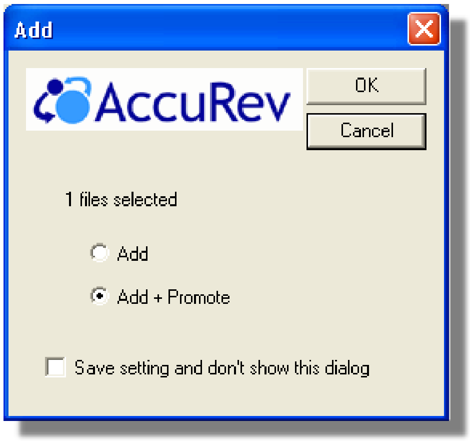


Add this project to AccuRev?

The IDEs can be configured to invoke this command automatically for a new project, or for a new file added to an existing project.

You can use an explicit command to place an entire project or individual file, under source control:

- Visual Basic: **Tools > AccuRev > Add Project to AccuRev**
- Visual C++: **Project > Source Control > Add to Source Control**
- Visual Studio .NET: **File > Source Control > Add Selected Projects to Source Control** (or **Add Solution to Source Control**)



1 files selected

○ Add

● Add + Promote

□ Save setting and don't show this dialog

The integration implements this command in either of these ways:

- Perform an **add** command, creating an initial version of each file in your workspace stream.
- Perform an **add** command, as above, and also perform a **promote** command to propagate the initial version of the file(s) to the backing stream.

Note: when Visual Basic adds a new project to source control, it handles the project file (**.vbp**) separately from the other files.

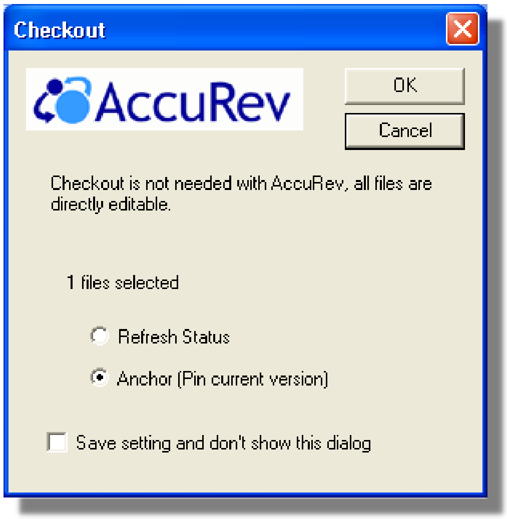
### Check Out

The SCC **Check Out** command is intended to make a file writable, so that you can edit it. With AccuRev, though, your files are *always* writable. So you don't need to perform **Check Out** commands when working in the IDE. AccuRev does have its own "check out" command —

called **co** (or equivalently, **anchor**). This doesn't change the writability of a file, but it does activate the file in your workspace (that is, adds it to the workspace's default group). One effect of this is to ensure that the file won't be overwritten by a **Get Latest Version** command (which performs an AccuRev **update** command). Under normal circumstances, you rarely need to invoke the **co** command.

Accordingly, when the integration performs a **Check Out**, there are two choices:

- Just have the IDE display a "checked out" icon for the file(s). (No change takes place at the AccuRev level.)
- Perform an AccuRev **co** (**anchor**) command, activating the file in your workspace.



Checkout is not needed with AccuRev, all files are directly editable.

1 files selected

○ Refresh Status

● Anchor (Pin current version)

□ Save setting and don't show this dialog

Note: turning on of "checked out" icons is also performed by command:

- Visual Basic: **Tools** > **AccuRev** > **Refresh File Status**
- Visual C++: **Project** > **Source Control** > **Refresh Status**
- Visual Studio .NET: **File** > **Source Control** > **Refresh Status**

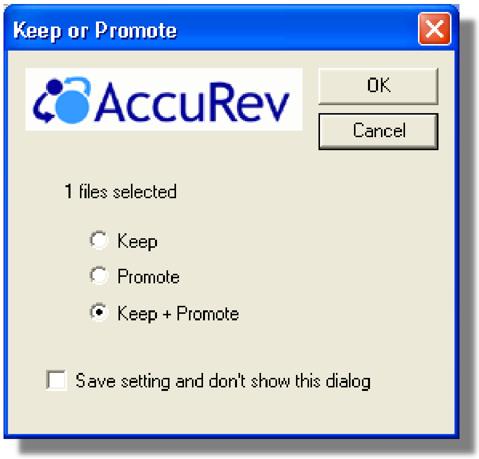
These commands examine all source files, and turn on the "checked out" icon for each one you've modified.

## Check In

The SCC **Check In** command saves the changes you've made to one or more files in the source-code repository. AccuRev's "two level check-in" model means you have the choice of keeping the changes private (in your workspace) or making them public (in the backing stream):

- Perform a **keep** command, creating a new version of each file in your workspace stream.
- Perform a **promote** command, propagating versions previously created with **keep** to the backing stream.
- Perform both **keep** and **promote** commands at the same time.

i



1 files selected

○ Keep

○ Promote

● Keep + Promote

□ Save setting and don't show this dialog

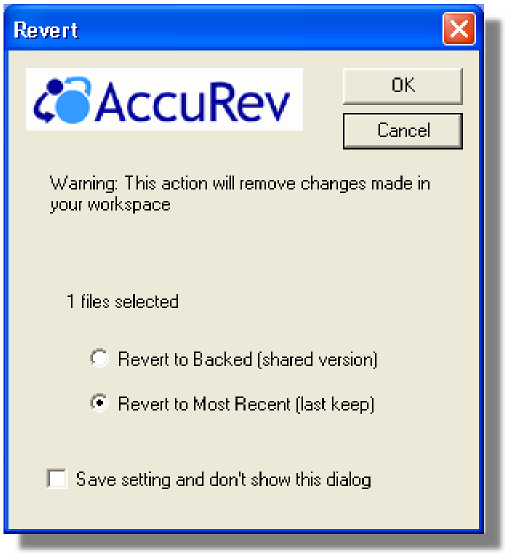
Note: the IDE removes the "checked out" icon from a file only when you promote it to the backing stream. If you execute the **keep**-only variant of the **Check In** command, the IDE still considers the file to be checked out.

## Undo Check Out

The SCC **Undo Checkout** command is intended to discard the changes you've made to a file after your most recent **Check Out**.

The integration implements this command in either of these ways:

- Discard all the changes you've made since the last time you performed an AccuRev **promote** of the file (with **Check In**). Note that this might discard several intermediate **keep** versions that you created with **Check In**.
- Discard just the changes you've made since the last time you performed an AccuRev **keep** of the file (with **Check In**).



Warning: This action will remove changes made in your workspace

1 files selected

○ Revert to Backed (shared version)

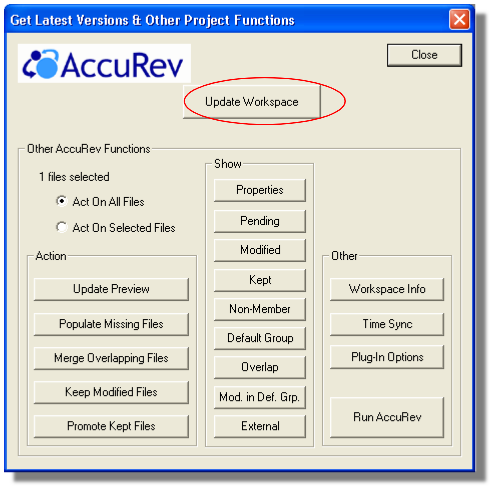
● Revert to Most Recent (last keep)

□ Save setting and don't show this dialog

## Get Latest Version

The SCC **Get Latest Version** command is intended to copy the most recent version of a file from the central repository to your workspace. AccuRev's **update** command does not operate on individual files, though — it updates the entire contents of your workspace. Thus, when you invoke this command, you may get more than you bargained for!

The AccuRev-specific dialog for the **Get Latest Version** command looks like this. To get the latest version (the version in the backing stream) of the specified file — and other files in your workspace, too — click the **Update Workspace** button.



Other AccuRev Functions

1 files selected

● Act On All Files

○ Act On Selected Files

The multitude of other buttons in this dialog provide access to a variety of AccuRev functions.

When you open an existing source-controlled project in Visual Basic, it can run this command automatically on all project files. (See AccuRev Plug-In Options on page 23.)

## Show Differences

The SCC **Show Differences** command compares your file with an older version in the repository. This command is located on the main menu (not on a file's context menu):

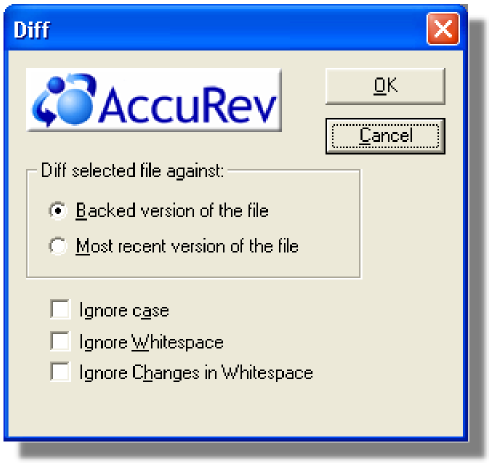
- Visual Basic: **Tools > AccuRev > Show Differences**
- Visual C++: **Project > Source Control > Show Differences**

The integration implements this command in either of these ways:

- Compare with the “public” version in the backing stream.
- Compare with the most recent “private” version you created in your workspace.

Either way, you have several options regarding the way case and whitespace are handled.

Note: only text files can be compared.


Diff

Diff selected file against:

● Backed version of the file

○ Most recent version of the file

□ Ignore case

□ Ignore Whitespace

□ Ignore Changes in Whitespace

## Show History

The SCC **Show History** command displays the set of transactions involving the selected file. This command is located on the main menu (not on a file’s context menu):

- Visual Basic: **Tools** > **AccuRev** > **Show History**. The history listing is sent to the AccuRev Messages window.
- Visual C++: **Project** > **Source Control** > **Show History**. The history listing is sent to the Source Control tab of the Output pane.
- Visual Studio .NET: **File** > **Source Control** > **History**. The history listing is sent to the AccuRev Messages window.

## Save As

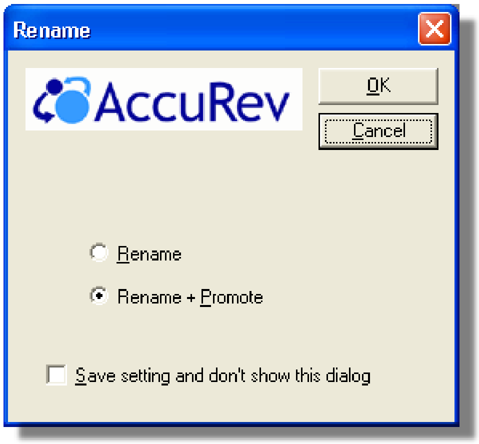
The IDE commands **Save As** and **Save Project As** create a new file. In Visual Basic, these commands trigger an SCC name-change operation, changing the name of the file in the repository. For example, if you select **Proj21.frm** and Save As to **widget_project.frm**, AccuRev uses the **move** command to rename file element **Proj21.frm** to **widget_project.frm**. The file **Proj21.frm** remains in your workspace, as a file not under source control (an “external” file, in AccuRev terms).

The integration implements this in either of these ways:

- Perform a AccuRev **move** command,
- Perform a AccuRev **move** command, followed by a **promote** of the element.

VS.NET has a **Rename** command (for files, projects, and solutions), but this command is not accompanied by an AccuRev **move** command. To avoid confusion, avoid using **Rename**.

VC++ does not have a **Save As** or **Rename** command.


Rename

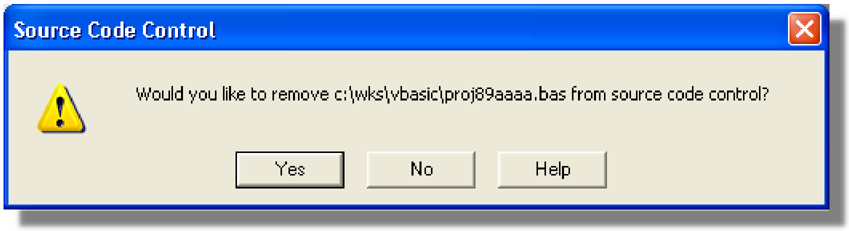
○ Rename

● Rename + Promote

□ Save setting and don't show this dialog

## Remove

The Visual Basic command **Remove <filename>** removes a file from a project, and also from your work area on disk. Visual Basic asks whether you also want to remove the file from the source-control system:



Source Code Control

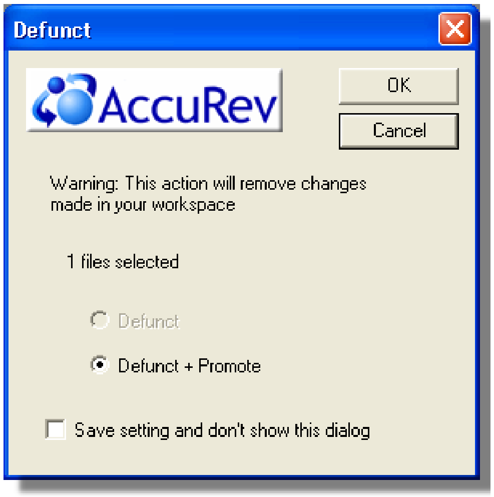
Would you like to remove c:\wks\vbasic\proj89aaaa.bas from source code control?

If you select **Yes**, the integration implements this in either of these ways:

- Perform a AccuRev **defunct** command (currently disabled).
- Perform a AccuRev **defunct** command, followed by a **promote** of the element.

n Visual C++, the command **Project > Source Control > Remove from Source Control** brings up the same AccuRev dialog.

VS.NET has a **Remove** command for projects and a **Delete** command for files. But these commands are not accompanied by AccuRev **defunct** commands. The data structure(s) disappear from the IDE, but remain under AccuRev source control.



Defunct

Warning: This action will remove changes made in your workspace

1 files selected

○ Defunct

● Defunct + Promote

□ Save setting and don't show this dialog

## Additional AccuRev-Related Commands

The AccuRev-SCC integration provides these additional commands. In Visual Basic, the commands are located under **Tools > Accurev**. In Visual C++, they're located under **Project > Source Control**.

### AccuRev Properties

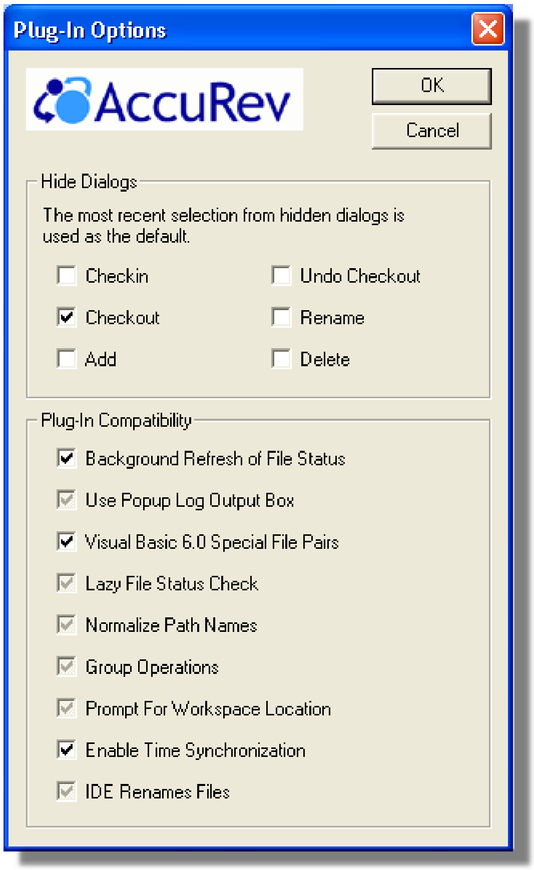
Sends the selected file's AccuRev status (**stat** command) to the messages/output window.

### Refresh Status

If you have modified the selected file without entering a **Check Out** command, redisplays the file with a "checked out" icon.

## AccuRev Plug-In Options

(Select **AccuRev** or **Run AccuRev** from the menu, then click **Plug-In Options**.) Displays a dialog with which you can control, on a command-by-command basis, the level of interactivity of AccuRev commands invoked through the SCC interface. Use the various checkboxes in the Hide Dialogs section to suppress the AccuRev dialogs invoked by particular SCC commands; the most recent selection you made in the dialog will be used automatically.



Hide Dialogs

The most recent selection from hidden dialogs is used as the default.

□ Checkin □ Undo Checkout
☑ Checkout □ Rename
□ Add □ Delete

Plug-In Compatibility

☑ Background Refresh of File Status
▣ Use Popup Log Output Box
☑ Visual Basic 6.0 Special File Pairs
▣ Lazy File Status Check
▣ Normalize Path Names
▣ Group Operations
▣ Prompt For Workspace Location
☑ Enable Time Synchronization
▣ IDE Renames Files

The checkboxes in the Plug-In Compatibility section control various aspects of AccuRev-SCC integration functionality.

> Note: these are tri-state checkboxes: the feature is either enabled (box is checked), disabled (box is cleared), or defaulted (box has gray check). “Defaulted” means to use the IDE’s default for the feature. (A particular IDEs might not support the feature at all.)

### Background Refresh of File Status

Controls periodic, automatic execution of the Refresh Status operation (see above).

### Use Popup Log Output Box

Controls use of an AccuRev-specific log window (“AccuRev Messages”), instead of the IDE’s own log window or tab. A change in this setting takes effect the next time you start the IDE.

### Visual Basic 6.0 Special File Pairs

Controls a facility in which certain files hidden by Visual Basic’s Project Explorer window are automatically processed at the same time as their visible “partner” files. For example, when you **Check In** the visible file **widgets.frm**, the hidden file **widgets.frx** will be checked in, too. This facility is initially enabled. The file pairs handled by this facility are:

| Visible File Suffix | Hidden File Suffix |
|---|---|
| .frm | .frx |
| .dob | .dox |
| .ctl | .ctx |
| .pag | .pgx |

**Lazy File Status Check**

Schedule automatic execution of the Refresh Status operation to occur less frequently (in batches, rather than for individual files). This improves performance, but increases the possibility that a file's status can be displayed incorrectly for a short time.

**Normalize Path Names**

(enable only if the depot is case-sensitive) Causes the integration to determine the proper case of each filename passed to it by the IDE.

**Group Operations**

Not yet implemented.

**Prompt for Workspace Location**

Enables a dialog in which you are prompted for the disk location of your workspace (in cases where the IDE doesn't supply this information).

**Enable Time Synchronization**

Enables/disables the **Time Sync** button.

**IDE Renames Files**

Tells the integration that the IDE itself can rename files at the operating-system level (checked), or tells the integration to do the renaming itself (not checked). This enables the AccuRev **move** command to proceed correctly, keeping a file's operating-system name in sync with its name in the AccuRev repository.

### Time Sync

Runs the AccuRev **synctime** command, to synchronize the system clock on your machine with the clock on the AccuRev Server machine.

### Run AccuRev

Starts the AccuRev GUI, enabling you to perform operations that are not supported by the AccuRev-SCC integration, such as file merging.

## Notes on IDE Usage

The following sections present some notes and suggestions regarding use of the various IDEs and their integrations with AccuRev.

## File Status Icons

The supported IDEs indicate the source-control status of files with special icons:

| | Visual Basic 6.0 | Visual C++ 6.0 | Visual Studio .NET |
|---|---|---|---|
| file not under source control | normal icon | white file icon | auxiliary red-checkmark icon to left of normal icon (pending checkin) |
| file under source control and checked-in | auxiliary file icon to left of normal icon | gray file icon | auxiliary blue-lock icon to left of normal icon |
| file under source control and read-only | auxiliary blue-lock icon to left of normal icon | | |
| file under source control and checked-out | auxiliary red-checkmark icon to left of normal icon | red checkmark to left of gray icon | auxiliary red-checkmark icon to left of normal icon |

## Files that Should Not Be Placed under Source Control

The IDEs create and manage various auxiliary files, which should not normally be placed under source control:

- Visual Basic 6.0: files with this suffix: **.dca**
- Visual C++ 6.0: files with these suffixes: **.ncb**, **.clw**, **.opt**

# Using AccuRev with Sun Java Studio

This white paper describes the integration of AccuRev with the Java-language IDEs NetBeans and Sun Java Studio. (Sun Java Studio, based on NetBeans, was formerly named “Sun ONE Studio” and “Forte for Java”.) The integration enables users of the IDE to access AccuRev version-control facilities using the IDE’s own “Versioning” menu.

As of AccuRev Version 3.5, this integration has been validated with:

- NetBeans 3.5
- Sun Java Studio Standard, Version 5

Check the AccuRev Release Notes for updated information on which IDEs the integration supports.

## Installing the JAR File

The AccuRev Integration is implemented as a single JAR (Java archive) file, named **ac_sunjava.jar**. The AccuRev installer places this file in the AccuRev **bin** directory.

## Enabling the Integration

Use this procedure to enable the AccuRev Integration in the Java IDE:

1. Start the Java IDE.
2. Select **Tools** > **Options** from the IDE menu.
3. In the Options window, navigate to **Options** > **IDE Configuration** > **System** > **Modules**.
4. Right-click on **Modules**, and select **Add** > **Module**.
5. In the file-selection dialog, navigate to the AccuRev **bin** folder, and select **ac_sunjava.jar**.

### Disabling the Integration

If you wish to disable use of the AccuRev Integration in the Java IDE:

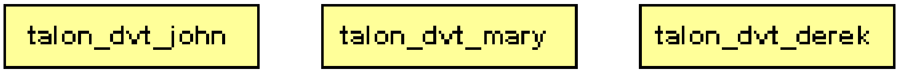
1. Select **Tools** > **Options** from the IDE menu.
2. In the Options window, navigate to **Options** > **IDE Configuration** > **System** > **Modules** > **Version Control** > **AccuRev**.
3. In the right side of the Options window, find the **Enabled** property, whose current value is **True**.
4. Click the **True** value, and use the list-box functionality to change the value to **False**.

# 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".

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 files. 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:

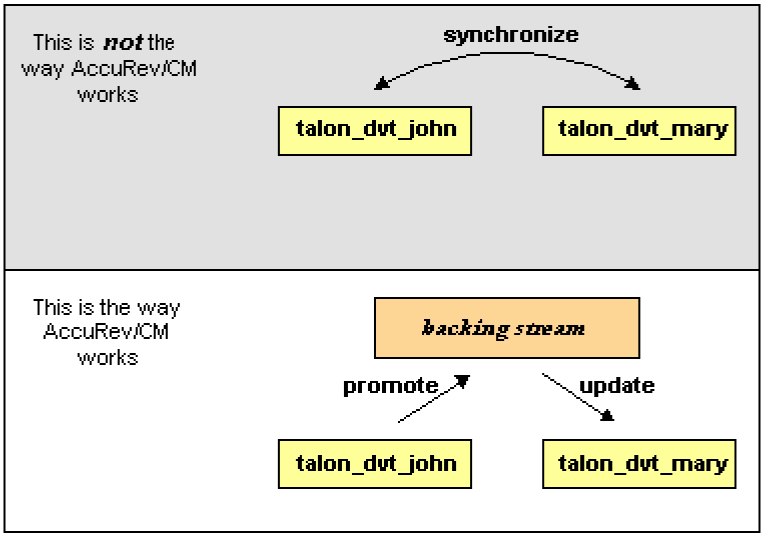


This set of users' workspaces uses the convention of having like names, suffixed with the individual usernames. AccuRev enforces this username-suffix convention. **talon_dvt** might mean "development work on the Talon product"; **john**, **mary**, and **derek** would be the users' operating system login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

From AccuRev's perspective, 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 files.
- The workspaces lose synchronization as each user makes changes to some of the files.
- Periodically, users share their changes with each other. When **john** incorporates some or all of **mary**'s changes into his workspace, their two workspaces become more closely (perhaps completely) synchronized.

You might assume that the workspace synchronization process involves the direct transfer of data from one workspace to another. But this is not the way AccuRev organizes the work environment. 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 promotion.
2. Whenever they wish, other team members incorporate the public changes into their own workspaces. This step is called updating.

The first step involves a public data area, called a stream. AccuRev has several kinds of streams; the kind that we're discussing here 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.

AccuRev also allows you to save any number of intermediate versions of a file in your workspace, before making your changes public. Such "private" versions of a file are created by the keep operation.

## Establishing Your Identity

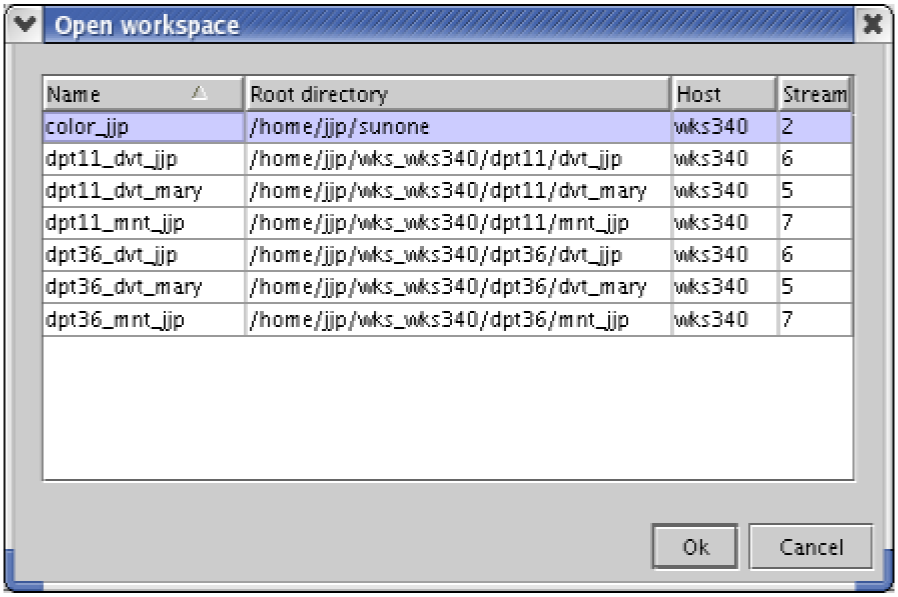
All AccuRev commands must be executed by a user who is listed in the AccuRev user registry. By default, the Integration uses your operating-system username as your AccuRev username. To have the Integration use a different AccuRev username, set environment variable ACCUREV_PRINCIPAL before starting the IDE. Here's a Bash Shell example:

```
export ACCUREV_PRINCIPAL=derekp
./runide.sh
```

## Using a Workspace

The various Integration commands move data between the central source-code repository (called a depot) and your personal work area (called a workspace). You must work in an existing AccuRev workspace. You must create this workspace with the AccuRev GUI or CLI; there is no Integration command to create a workspace.

To work with the files in a particular workspace, you must "mount" it. (This is the IDE's terminology; it's different from the basic Unix/Linux concept of making a disk partition available as a file system.) The command **Versioning > Mount AccuRev workspace** displays a list of all workspaces:

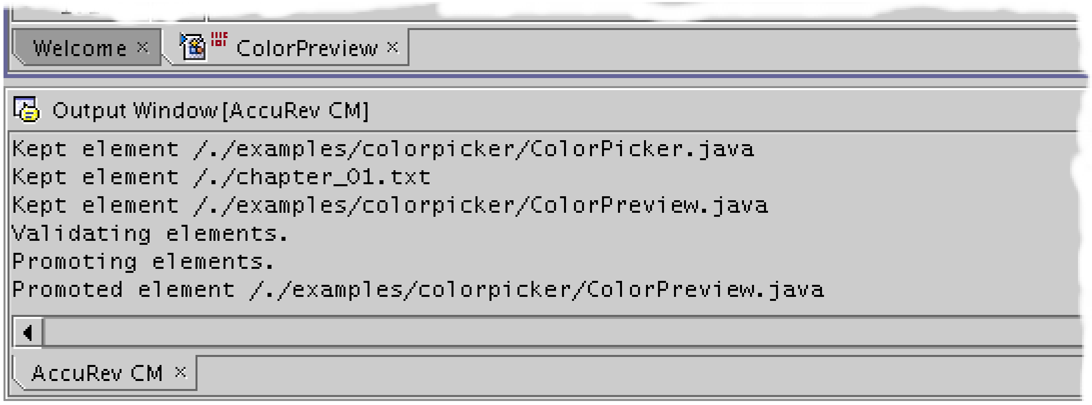


| Name | Root directory | Host | Stream |
|---|---|---|---|
| color_jjp | /home/jjp/sunone | wks340 | 2 |
| dpt11_dvt_jjp | /home/jjp/wks_wks340/dpt11/dvt_jjp | wks340 | 6 |
| dpt11_dvt_mary | /home/jjp/wks_wks340/dpt11/dvt_mary | wks340 | 5 |
| dpt11_mnt_jjp | /home/jjp/wks_wks340/dpt11/mnt_jjp | wks340 | 7 |
| dpt36_dvt_jjp | /home/jjp/wks_wks340/dpt36/dvt_jjp | wks340 | 6 |
| dpt36_dvt_mary | /home/jjp/wks_wks340/dpt36/dvt_mary | wks340 | 5 |
| dpt36_mnt_jjp | /home/jjp/wks_wks340/dpt36/mnt_jjp | wks340 | 7 |

To mount a workspace in the IDE, select it and click **Ok**.

## AccuRev Command Output

AccuRev commands generate confirmation messages, which are displayed within the IDE's Output Window pane, in a tab labeled **AccuRev CM**.



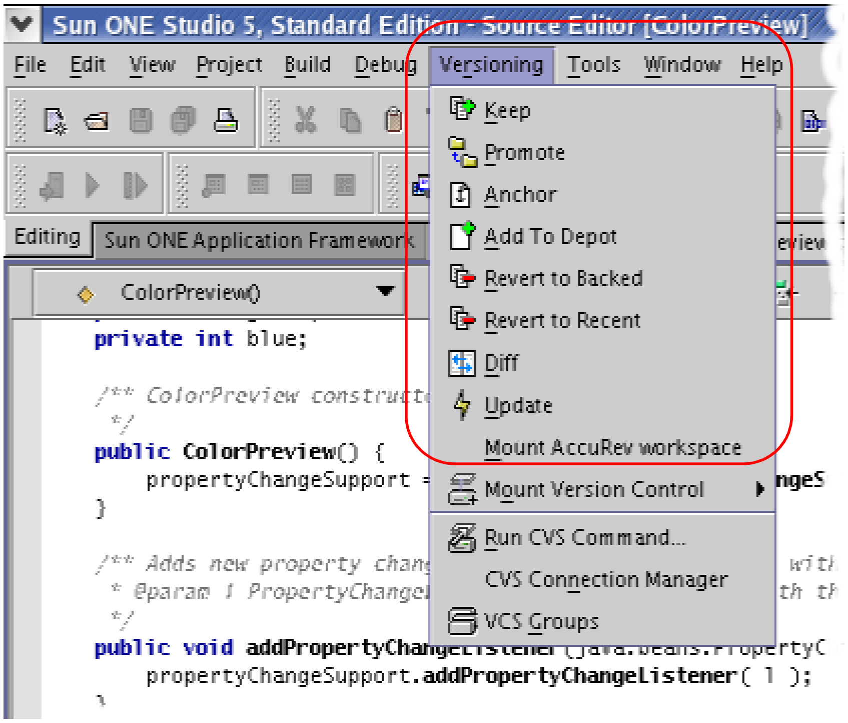
```
Kept element /./examples/colorpicker/ColorPicker.java
Kept element /./chapter_01.txt
Kept element /./examples/colorpicker/ColorPreview.java
Validating elements.
Promoting elements.
Promoted element /./examples/colorpicker/ColorPreview.java
```

## Command Summary

All AccuRev commands supported by the Integration appear in the IDE’s **Versioning** menu. These commands operate on the file currently displayed in the IDE’s Editing pane.

The same set of AccuRev commands also appear in the context menu (right-click) for files in the Filesystem Explorer. These commands operate on the currently selected file(s) in the Explorer pane.

The AccuRev commands supported by the Integration are described in the sections below.



```
private int blue;

/** ColorPreview constructo
 */
public ColorPreview() {
    propertyChangeSupport =
}

/** Adds new property chang
 * @param l PropertyChange
 */
public void addPropertyChange
    propertyChangeSupport.addPropertyChangeListener( l );
}
```

### Keep

The **Keep** command 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 “public” backing stream or in other users’ workspaces.

### Promote

The **Promote** command 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.

### Anchor

Many version control systems have a “check out” command that makes a file writable, so that you can edit it. But with AccuRev, your files are *always* writable. AccuRev *does* have its own similar command, called **Anchor**, This doesn’t change the writability of a file, but it does activate the file in your workspace (that is, adds it to the workspace’s default group). One effect of this is to ensure that the file won’t be overwritten by an **Update** command. Under normal circumstances, you rarely need to invoke the **Anchor** command.

## Add to Depot

The **Add to Depot** command converts one or more of the files in the development project into AccuRev version-controlled elements. (The directory containing the files is also converted to an element, if necessary.) The files to be converted must be located in a directory within an existing AccuRev workspace.

## Revert to Backed

The **Revert to Backed** command discards the changes you've made to a file. It restores the version that was in the backing stream at the time of your most recent **Update**. (But if you promoted one or more versions of the element to the backing stream since your most recent **Update**, it restores the most recently promoted version.)

## Revert to Recent

The **Revert to Recent** command is similar to **Revert to Backed**, but rolls back a file only as far as the private version you recently created with **Keep**. This command is useful when you edit a file, save it with **File > Save**, then decide to throw away the changes.

## Diff

The **Diff** command compares your file with the private version you recently created with **Keep**. ("What have I changed since my last **Keep**?")

> Note: only text files can be compared.

## Update

The **Update** command 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.