Development Environment Administration Guide

VERSION 2018.0.01

jade

Contents

Contents iv

Before You Begin

The *JADE Development Environment Administration Guide* is intended as the main source of information when you are administering the JADE development environment.

Who Should Read this Guide

The main audience for the *JADE Development Environment Administration Guide* is expected to be system administrators.

What's Included in this Guide

The *Development Environment Administration Guide* has three chapters.

Chapter 1	Covers saving and restoring schema and forms definitions
Chapter 2	Covers change control
Chapter 3	Covers patch versioning

Related Documentation

Other documents that are referred to in this guide, or that may be helpful, are listed in the following table, with an indication of the JADE operation or tasks to which they relate.

Title	Related to…
JADE Encyclopaedia of Classes	System classes (Volumes 1 and 2), Window classes
JADE Encyclopaedia of Primitive Types	Primitive types and global constants
JADE Developer's Reference	Developing or maintaining JADE applications
JADE Installation and Configuration Guide	Installing and configuring JADE
JADE Initialization File Reference	Maintaining JADE initialization file parameter values
JADE Database Administration Guide	Administering JADE databases
JADE Synchronized Database Service (SDS) Administration Guide	Administering JADE Synchronized Database Services (SDS), including Relational Population Services (RPS)
JADE Object Manager Guide	JADE Object Manager administration, including security
JADE Report Writer User's Guide	Using the JADE Report Writer to develop and run reports
JADE Development Environment User's Guide	Using the JADE development environment

Conventions

The *JADE Development Environment Administration Guide* uses consistent typographic conventions throughout.

Convention	Description
Arrow bullet (❯❯)	Step-by-step procedures. You can complete procedural instructions by using either the mouse or the keyboard.
Bold	Items that must be typed exactly as shown. For example, if instructed to type **foreach**, type all of the bold characters exactly as they are printed. File, class, primitive type, method, and property names, menu commands, and dialog controls are also shown in bold type, as well as literal values stored, tested for, and sent by JADE instructions.
Italic	Parameter values or placeholders for information that must be provided; for example, if instructed to enter *class-name*, type the actual name of the class instead of the word or words shown in italic type. Italic type also signals a new term. An explanation accompanies the italicized type. Document titles and status and error messages are also shown in italic type.
Blue text	Enables you to click anywhere on the cross-reference text (the cursor symbol changes from an open hand to a hand with the index finger extended) to take you straight to that topic. For example, click on the "Patch Versioning" cross-reference to display that topic.
Bracket symbols ([ ])	Indicate optional items.
Vertical bar (|)	Separates alternative items.
`Monospaced font`	Syntax, code examples, and error and status message text.
ALL CAPITALS	Directory names, commands, and acronyms.
SMALL CAPITALS	Keyboard keys.

Key combinations and key sequences appear as follows.

Convention	Description
KEY1+KEY2	Press and hold down the first key and then press the second key. For example, "press Shift+F2" means to press and hold down the Shift key and press the F2 key. Then release both keys.
KEY1,KEY2	Press and release the first key, then press and release the second key. For example, "press Alt+F,X" means to hold down the Alt key, press the F key, and then release both keys before pressing and releasing the X key.

Chapter 1 Administering Your JADE Development Environment

This chapter covers the following topics.

Saving Your JADE Schema and Forms Definitions

Use the JADE extract options to tailor the saving of your current schema or to extract a specific class, interface, or method.

Note You can extract only your user-defined schemas. You cannot extract or load the **RootSchema** (the base schema that contains system objects).

The **Extract** command from the Schema menu enables you to save (extract) the current schema to a file.

Use the Schema menu **Extract** command when you are:

- Backing up a full or partial schema
- Passing code to another user
- Fully reconstructing a schema in the same JADE release

The Schema menu **Extract** command enables you to extract:

- A full or partial schema
- Forms from all applications in a subschema

The Classes, Interfaces, and Methods menus provide facilities to extract a selected method, interface, class, or class and its subclasses, and write them to a file. Extracting a class, interface, or method provides you with a quick means of creating an extracted file of a class, interface, or method; for example, for a backup, to pass the code to another user, or to reconstruct.

Tip Using this command is a simpler, faster way to extract an individual method, interface, class, or class and subclasses than performing a selective extract of your schema.

When you extract a class, its associated methods, properties, and so on, are also extracted. Extracting an interface also extracts its associated constants and methods. For more details, see "Extracting Your Schema", in Chapter 10 of the *JADE Development Environment User's Guide*.

Restoring Your JADE Schema and Forms Definitions

Use the **Load** command from the Schema menu to load your schema, forms, or specific class, interface, or method from an extract file.

Note You can load only your extracted user-defined schemas. You cannot extract the **RootSchema** (the base schema that contains system objects).

The **Load** command from the Schema menu enables you to load (install) a full or partial schema from an extracted file into your current schema.

Use the Schema menu **Load** command when you are:

- Converting between JADE releases
- Restoring a full or partial schema
- Passing code to another user
- Fully reconstructing a schema in the same JADE release

The Schema menu **Load** command enables you to load:

- A full or partial schema
- Forms from all applications in a subschema

For more details, see "Loading Your Schema", in Chapter 10 of the *JADE Development Environment User's Guide*.

Change Control and Patch Versioning Considerations

When using change control or patch versioning:

- Use patch versioning to control historical data (that is, each change to any entity such as classes, interfaces, methods, properties, or constants) in the JADE development environment.

  The latest patch control version number applies to a method checked out by you to a delta. A method checked out by another user relates to the penultimate patch version number.

- Deltas and the associated change control relate only to methods, and are useful for runtime testing. Change control applies only to methods, which are checked into and out of deltas.

  A delta applies only to the latest patch control version. It does not maintain historical information about changes to methods.

- When patch versioning is enabled, you can execute only the latest version of a method. (The changes that you can view by using the Compare Sources window accessed from the Summary of Patches window are for display purposes only, although you can apply a specific version of a source to become the latest source.)

  When using change control and you are not set to the delta in which a method is checked out, you can execute the previous source of the checked out method; that is, the checked in version.

- Checking out a method enables you to test that method without affecting other users who are not set to that delta.

  Any testing of a changed method when patch versioning is enabled affects all other developers, regardless of any delta in which they may be working.

- A method is checked out at the delta level and not at the level of the user within that delta who checks out the method.

  If patch versioning is set, any developer who has access to that method can see all method source changes.

- When you have patch versioning enabled for a schema and a method is checked out in another delta, the Summary of Patches window displays only the method that is checked in.

You cannot view (or compare) any changes made to that checked out method until it has been checked back in to the delta.

- Extracting changed entities extracts only the latest change. For example, if five changes have been made to a method source in one patch version, only the last change is extracted; that is, earlier source changes in that patch version are lost.

  Tip You can, however, use the **Apply** button in the Compare Sources window (accessed from the Summary of Patches window) to compare change versions to a source within a patch and to apply a specific change so that it becomes the latest one before you extract your changes. However, applying an earlier change to a method source may affect other developers.

  When extracting from a delta in which methods are checked out, the checked out versions of those methods are extracted. However, checked *in* methods are extracted if you are extracting from another delta; that is, your extracted method sources may not be the latest.

For details, see "Change Control" in Chapter 2 and "Patch Versioning" in Chapter 3.

Chapter 2 Change Control

This chapter covers the following topics.

Overview

To enable a user to check methods into and out of the JADE database, a delta must be set.

Deltas apply to all schemas in your JADE development environment, including the **RootSchema** and any other superschema or subschema, to enable more than one developer to work in the same delta, if required.

You can specify a delta code when extracting a schema from a parameter file; for example, to automate the checking in of a delta at run time so that you can create the release of a change for testing purposes. For details about automating the extraction of a schema, see "Extracting Schemas as a Non-GUI Client Application", in Chapter 10 of the *JADE Development Environment User's Guide*.

You can specify the optional **delta** command line argument when running a non-GUI client application using the **jadclient** executable. For details, see "Running a Non-GUI Client Application using jadclient", in Chapter 1 of the *JADE Runtime Administration Guide*.

In a multiuser JADE environment, deltas enable developers working in different parts of JADE to lock methods so that they cannot be accessed by developers in other parts of the JADE development environment. (See also "Change Control and Patch Versioning Considerations", in Chapter 1.)

Deltas are also useful if you want to test the modification of a method and are unsure if you want to keep your modifications. Checking out a method, making your changes, and then selecting the **Undo Checkout** command from the Methods menu Delta submenu in the Class Browser or Primitive Types Browser enables you to discard your changes.

Notes Methods can be checked in and out only when a delta is set.

If you want methods to be checked out only to the current delta instead of the default action if methods being checked out to multiple deltas, set the **SingleDelta** parameter in the [DeltaOptions] section of the JADE initialization file to **true**. Setting the parameter to **true** is ignored for methods that have already been checked out to multiple deltas; it is effective only for all methods that are subsequently checked out.

The **Remove** command in the Methods menu is disabled when a method from an exported interface is checked out, so that neither the original method nor the checked out method or methods can be removed. Before you can remove the method, you must check it in or undo the checking out of the method.

When a user is set to a delta, checked out versions of methods are extracted when a schema is extracted.

You can change your default delta options, if required, by using the **Source Management** sheet from the Options menu **Preferences** command. From the Schema Browser, select (and set, if required) the schema whose deltas you want to view or maintain.

To open a Delta Browser window

- Select the **Deltas** command from the Browse menu.

A Delta Browser window is then opened. If you have not yet defined a delta, nothing is displayed in the Delta Browser.

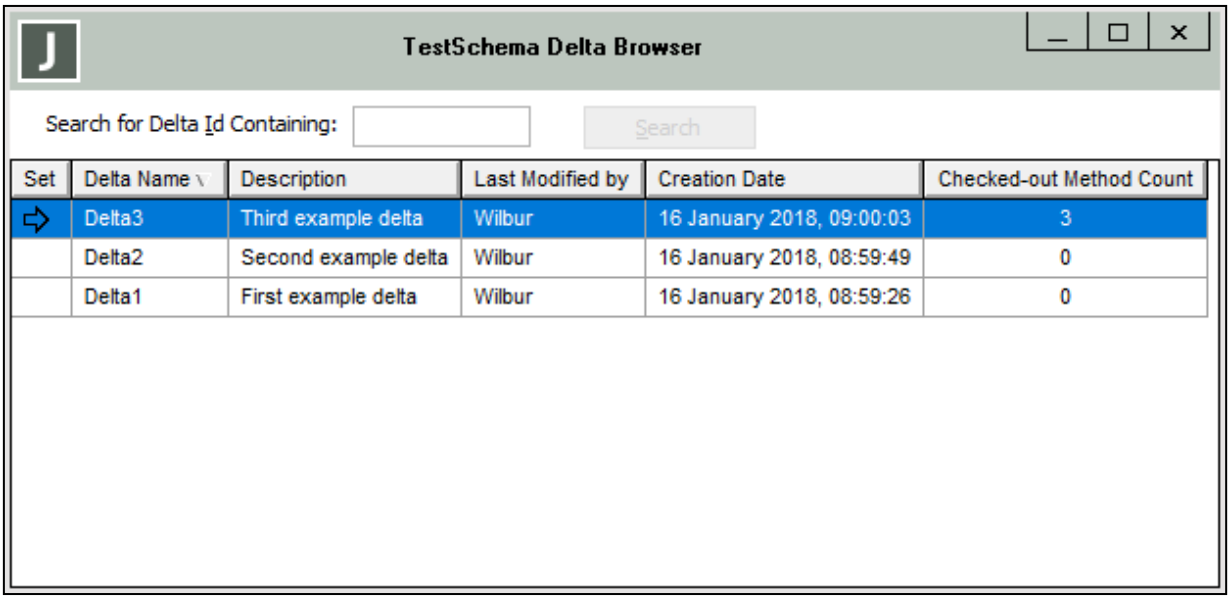
Only one Delta Browser for the current schema can be open at any time. If a Delta Browser is already open for that schema, it is brought to the top when you select the **Deltas** command from the Browse menu.

You can have concurrent open Delta Browsers for different schemas in a development session.

Maintaining Deltas

The Delta Browser, accessed from the **Deltas** command in the Browse menu, enables you to maintain deltas.

The following image is an example of the Delta Browser.



The Delta Browser provides a summary of delta attributes, which are displayed in a table that has the following columns.

1. **Set**, with an arrow in the row indicating the currently set delta. This column is empty if no delta has been set.
2. **Delta Name**, containing the delta identifier.
3. **Description**, containing the delta description.
4. **Last Modified by**, containing the name of the user who created or last updated the delta information.
5. **Creation Date**, containing the creation timestamp of the delta.
6. **Checked-out Method Count**, containing the number of checked out methods for the delta.

Tip Right-click in the table on the Delta Browser to display the Delta menu commands.

You can sort the delta table by clicking on a column in the fixed row of the table. Clicking the same column again toggles between ascending and descending sort order for the **sort** property. By default, the list is displayed in **Delta Name** ascending order.

The count of the number of checked out methods displayed is updated when a method is checked out, unchecked out, or checked in.

To search for text in a delta

1. Enter the text in the **Search for Delta Id Containing:** text box.
2. Click the **Search** button.

The next delta, starting from the delta after the currently selected delta, that contains the specified text (which is case-insensitive), is then selected.

When the end of the table entries is passed, the search wraps around and starts at the first entry. If no delta contains that text or the only delta found is the currently selected delta, the display remains unchanged.

The Delta menu, accessed from the Delta Browser, contains the commands listed in the following table.

Command	Action	For details, see…
Add	Displays the Add Delta dialog	Adding a Delta
Change	Displays the Change Delta Definition dialog	Changing a Delta
Close	Checks in all checked out methods and deletes the selected delta	Closing a Delta
View Methods	Displays the Checked Out Methods Browser for the selected delta	Viewing Methods Checked Out to a Delta
Check In All	Checks in all methods checked out to the selected delta	Checking In All Methods
Undo Check Out All	Discards any changes in all checked out methods	Discarding Checked Out Method Changes
Set	Sets the selected delta to the current delta	Setting a Delta
Unset	Closes all open source windows, and unsets the selected delta	Unsetting a Delta

For details, see the following subsections.

Adding a Delta

From the Delta Browser, use the **Add** command from the Delta menu to add a delta.

When you load a schema that contained delta information when it was extracted, you must redefine your deltas in the newly loaded schema, as delta information is not extracted or loaded.

If you want checked out methods loaded, check the **Check Out Methods** check box on the Advanced Load Options dialog. (For details, see "Specifying Advanced Load Options", in Chapter 10 of the *JADE Development Environment User's Guide*.)

» To add a delta

1. Select the **Add** command from the Delta menu. The Add Delta dialog is then displayed.
2. In the **Delta Id** text box, enter an identifier name for your delta.

   The delta name must be unique, and cannot contain spaces. JADE converts the first character to uppercase.
3. In the **Description** text box, enter a description of the delta, if required. The description can be used to uniquely identify a delta.
4. Click the **OK** button when you have defined your new delta.

   Alternatively, click the **Cancel** button to abandon your selections.

When you click the **OK** button, the Delta Browser then displays your newly defined delta.

Changing a Delta

From the Delta Browser, use the **Change** command from the Delta menu to change an existing delta selected in the browser. (When you load a schema that contained delta information when it was extracted, you must redefine your deltas in the newly loaded schema, as delta information is not extracted or loaded. For details, see "Adding a Delta".)

Note If patch versioning is used, changing the name of a delta also updates the name of the delta in any JADE Patch Version detail entries for that delta.

To change a delta

1. Select the **Change** command from the Delta menu. The Change Delta Definition dialog is then displayed.
2. In the **Delta Id** text box, enter the new identifier name for the delta.

   The delta name must be unique, and cannot contain spaces. JADE converts the first character to uppercase.
3. In the **Description** text box, change the description of the delta, if required. The description can be used to uniquely identify a delta.
4. Click the **OK** button when you have maintained your delta. Alternatively, click the **Cancel** button to abandon your selections.

When you click the **OK** button, the Delta Browser then displays your updated delta details.

Closing a Delta

From the Delta Browser, use the **Close** command from the Delta menu to close (remove) the selected delta.

When you close a delta that contains checked out methods, you are prompted to confirm that all checked out methods are to be checked in before the delta information is removed.

A delta can be closed only when all methods that were checked out to that delta are checked in and when no other users are set to that delta.

To close a delta

1. Select the **Close** command from the Delta menu.
2. If methods are currently checked out to the selected delta, a Confirm dialog prompts you to confirm that you want to check the methods in and then close the delta. Click the **OK** button to confirm that you want to do this or click the **Cancel** button to abandon the closure.

   If no methods are checked out to the selected delta, a Confirm dialog advises you that there are no methods in the selected delta and prompts you to confirm that you want to close the delta. Click the **OK** button to confirm that you want to close the delta. Alternatively, click the **Cancel** button to abandon the closure.

When you click the **OK** button, any methods checked out to the selected delta are then checked in and focus returns to the Delta Browser, which no longer contains the closed delta.

Viewing Methods Checked Out to a Delta

From the Delta Browser, use the **View Methods** command from the Delta menu to view the methods checked out to a delta.

The **View Methods** command is disabled if no methods are currently checked out to the selected delta.

To view methods checked out to the selected delta

- Select the **View Methods** command from the Delta menu.

The Checked Out Methods Browser for the selected delta is then displayed. Use this window to Browser to view all methods that are currently checked out to the selected delta.

The method list displays in alphabetical order by class all methods that are currently checked out to the selected delta.

To view the logic for a checked out method

- Select the required method in the method list.

The logic for the selected method is then displayed in the editor pane.

The status line displays the access, creator of the method, creation timestamp, and the status of the method (that is, it is checked out); for example:

```
Access is public. Created by wilbur (7.1.10) on 31 August 2016, 10:07:42 (Checked
out)
```

You can modify and compile user-defined JADE methods in the editor pane of the Checked Out Methods Browser.

Checking In All Methods

From the Delta Browser, use the **Check In All** command from the Delta menu to check in all methods that are currently checked out to the selected delta.

When you use the **Check In All** command to check in all methods in the selected delta, all methods are checked in but the delta is not closed (removed).

The check in process compiles any method that you have modified but not compiled. Methods that contain errors are checked in but are marked as *"In error"*.

The **Check In All** command is disabled if no methods are currently checked out to the selected delta.

To check in all methods from the selected delta

1. Select the **Check In All** command from the Delta menu.
2. A Confirm dialog advises you of the number of methods that are currently checked out to the selected delta and prompts you to confirm that you want to check in all methods in the delta.
3. Click the **OK** button to confirm that you want to check in all methods.

   Alternatively, click the **Cancel** button to abandon the checking in.

When you click the **OK** button, all methods checked out to the selected delta are then checked in and focus then returns to the Delta Browser.

Discarding Checked Out Method Changes

From the Delta Browser, use the **Undo Check Out All** command from the Delta menu to discard any changes that have been made to all methods checked out in the delta.

When you use the **Undo Check Out All** command to discard all changes to methods in the delta, all changes to methods since the methods were checked out are discarded and the methods are checked back in, but the delta is not closed (removed).

The **Undo Check Out All** command is disabled if no methods are currently checked out to the selected delta.

To discard any changes to all methods in the delta

1. Select the **Undo Check Out All** command from the Delta menu.
2. A Confirm dialog advises you of the number of methods that are currently checked out to the selected delta and prompts you to confirm that you want to undo the changes to those methods in the delta.
3. Click the **OK** button to confirm that you want to undo the check out of all methods.

   Alternatively, click the **Cancel** button to abandon the checking in.

When you click the **OK** button, any changes that have been made by you or by another user of checked out methods are then discarded and the methods are checked in.

The status line no longer states that a method selected in the Methods List is checked out, and the **Undo Check Out All** command is then disabled.

Setting a Delta

From the Delta Browser, the **Set** command from the Delta menu enables you to set the selected delta to the current delta.

The **Set** command is enabled only when you select a delta that is not the current delta.

Note Methods that are checked out are checked out to the delta that is currently set. If no delta is set as the current delta, no methods can be checked in or out.

To set a delta as the current delta

1. In the Delta Browser, select the delta that you want to set as the current delta.
2. Select the **Set** command from the Delta menu.

Tip Double-click on a delta in the Delta Browser to quickly set it to the current delta.

A gray arrow is then displayed in the **Set** column of the selected delta in the Delta Browser, indicating that it is the current delta. That delta remains the current delta until another is set.

When a delta is set, the delta text display on the right of the JADE development environment toolbar is drawn in red. When you double-click on this delta text description, the Delta Browser is displayed.

Unsetting a Delta

From the Delta Browser, the **Unset** command from the Delta menu enables you to unset the delta that is currently set. When there is no delta set, no methods can be checked in or out.

The **Unset** command is enabled only when you select the delta that is the current delta.

To unset a delta

1. In the Delta Browser, select the current delta.
2. Select the **Unset** command from the Delta menu.

Tip You can also unset a delta by double-clicking on the currently set delta entry in the table in the Delta Browser.

When you unset the current delta, any methods that are checked out are checked in when the delta is unset. The gray arrow is then removed from the **Set** column of that delta in the Delta Browser.

No methods can be checked out until you set another delta as the current delta.

Maintaining Methods Checked In and Out

The Delta submenu in the Methods menu from the Class Browser or Primitive Types Browser contains the commands listed in the following table, to enable you to maintain the checking in or out of a method selected in the Methods List.

Command	Action	For details, see…
Check Out	Displays the CheckOut Options dialog	Checking Out a Method
Check In	Checks in the selected method	Checking In a Method
Compare	Compares the original and the updated method source	Comparing a Method Source
Undo Checkout	Checks in the selected method and discards any changes	Discarding Changes Made to a Checked Out Method

For details, see the following subsections.

In addition, the Methods menu provides the **Extract** command, which displays the standard Save As dialog. For details, see "Extracting Changed Methods".

Checking Out a Method

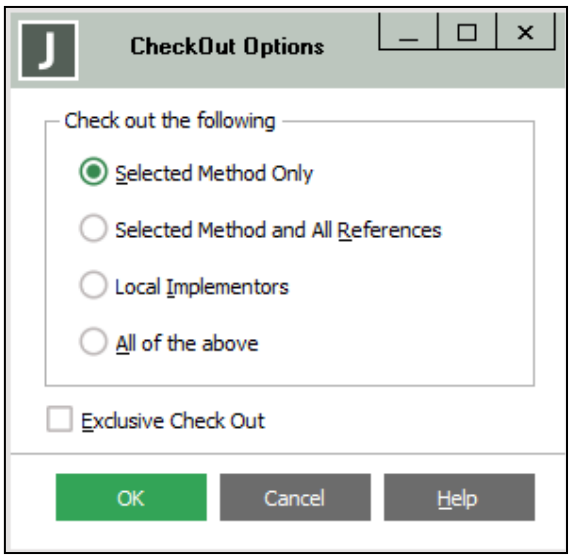
Use the **Check Out** command from the Methods menu Delta submenu in the Class Browser or Primitive Types Browser to check out the method selected in the Methods List.

Note Methods can be checked out only when a delta is set.

To check out a method

1. In the Methods List of the Class Browser or Primitive Types Browser, select the method that you want to check out.

2. Select the **Check Out** command from the Methods menu Delta submenu. The CheckOut Options dialog, shown in the following image, is then displayed.



3. If you do not want to check out only the selected method, select the required option; that is, check out the selected method and all references, local implementors, or both references and local implementors.
4. If you want to check out the method so that it cannot be updated by other users who are set to the current delta, check the **Exclusive Check Out** check box.

   By default, any user set to the current delta can update a checked out method.
5. Click the **OK** button. Alternatively, click the **Cancel** button to abandon your selection.

The selected method (and references or local implementors, if applicable) is then checked out. The status line states that the method is checked out and the method is locked to all other developers until you check it back in.

Note When a method is checked out, more than one developer can access that method by setting to the same delta.

The **Check Out** command is then disabled for the selected method until the method is checked in or the **Undo Checkout** command is selected.

Checking In a Method

Use the **Check In** command from the Methods menu Delta submenu in the Class Browser or Primitive Types Browser to check in the method selected in the Methods List.

The **Check In** command is enabled only when the selected method is checked out and a delta is set.

Methods that have changed but are uncompiled are compiled when they are checked in, and methods that are in error are marked accordingly.

» To check in a method

1. In the Methods List of the Class Browser or Primitive Types Browser, select the method that you want to check in.
2. Select the **Check In** command from the Methods menu Delta submenu.

A Confirm dialog advises you if the method source has not changed and prompts you to click the **OK** button to confirm that you want to check the method in.

The selected method is then checked in, the status line no longer states that the method is checked out, and the **Check In** command is disabled.

Tip You can check in all methods that are currently checked out to a delta, by selecting the **Check In All** command from the Delta menu in the Schema Browser.

Comparing a Method Source

Use the **Compare** command from the Methods menu Delta submenu in the Class Browser or Primitive Types Browser to compare the original and updated sources of the selected checked out method.

Note Method sources can be compared only when a delta is set and the method is checked out.

You can change your default source comparison options, if required, by using the Compare Sources group box controls of the **Source Management** sheet from the Options menu **Preferences** command. For example, you can select the comparison of changes only or that white space is to be compared.

To compare the original and updated sources of a checked out method

1. In the Methods List of the Class Browser or Primitive Types Browser, select the checked out method whose sources you want to compare.
2. Select the **Compare** command from the Methods menu Delta submenu.

3. The Compare Sources window, shown in the following image, is then displayed.

The upper part of the Compare Sources window (for the default **Visual** option in the Preferences dialog **Source Management** sheet) contains two panes, displaying the original source code of the method and the updated source code of the checked out method. As these panes are read-only, you cannot change the source code in either of them.

By default, the pane at the left of the window displays the source that is currently checked out and the pane at the right of the window displays the method source in the delta that is currently set. If you want to view the original source (that is, before the method was checked out to a delta), select the **Original** value in a combo box above the left or the right pane. Use these to compare the checked out method with the current source and the original source.

The merge editor pane in the lower area of the Compare Sources window enables you to copy source code from the original or updated source code pane, update it to meet your requirements, and then save it to your current delta.

When you have multiple deltas, the original source may not be the same as the current source, because a method can be changed in more than once.

If a method has been checked out to more than one delta, the combo boxes enable you to select the delta in which you want to compare the method source.

The comparisons that are displayed in the Compare Sources window are listed in the following table.

Color	Description
Blue	Addition in method line
Green	Change in method line
Red	Deletion in method line

The popup menu accessed from either of the source code panes provides the **Copy to Merge Window** command, which copies all of the code in that pane to the merge editor pane. You can then change the copied source and then save it by clicking the **Save** button. The saved source is saved to your *current* delta.

Tips By default, changes to white space (for example, a space or a tab) are ignored. If the **Ignore White Space** option is checked in the **Source Management** sheet of the User Preferences dialog, it may appear that no changes have been made even though the source has been modified; for example, indentation may have been changed.

You can copy code to the clipboard from either the current or the latest source windows displayed in the Compare Sources window, by using the **Copy** command in the Edit menu (or press Ctrl+C) or the **Copy** toolbar button so that you can then paste the code into a text editor or into the editor pane of another method, for example. The source code window that is copied is the one that has focus.

Alternatively, you can click the **Next** button to highlight the next line of your source code that has been updated and the original source, if applicable, or the **Previous** button to highlight the previous line of source code that has been updated and the original source.

Merging Compared Source

You can update source code copied from the original or the updated source code pane and then save it to your current delta.

To merge compared source code

1. Right-click in the original or the updated source code pane of the Compare Sources dialog and then select the **Copy to Merge Window** command from the popup menu that is displayed.

   All code in that source code pane is then copied to the lower merge editor pane.

2. Make any changes that you require to the source code copied to the merge editor pane.

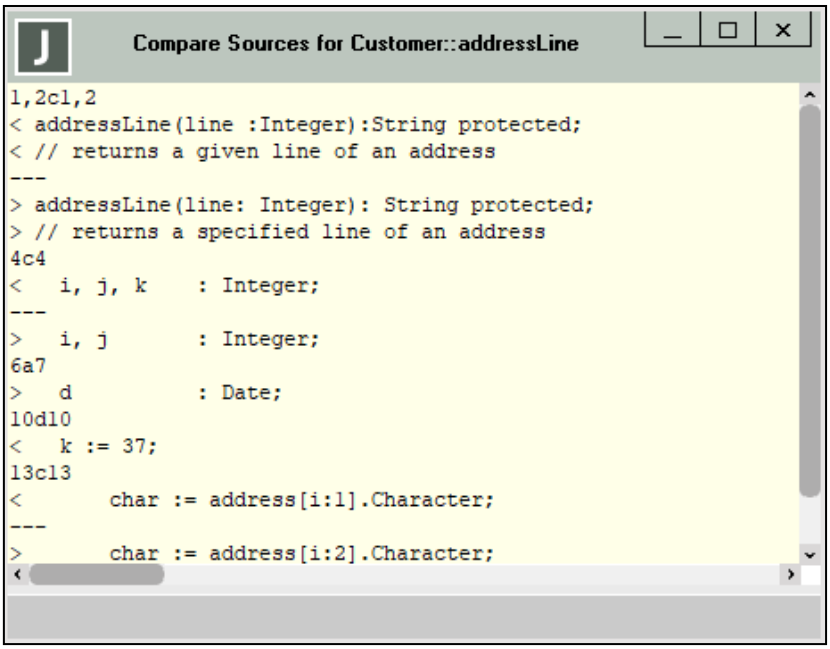
3. Click the **Save** button.

The saved source is saved to your *current* delta.

Viewing the Display of Changes to a Checked Out Method

You can view only the changes made to the source of a checked out method, by selecting the **Changes Only** option button in the View Options group box of the **Source Management** sheet, accessed from the Options menu **Preferences** command.

The Compare Sources window, shown in the following image using the same changes as those in the previous image, is then displayed.



In general, source modifications are displayed in the following format.

```
old-source indicator new-source
```

The additions, changes, and deletions are displayed in blue, green, and red, respectively, with locators **a**, **c**, or **d** representing the type of change made for a specific line number. For example:

Value	Description
4c4	Line 4 in *old-source* has changed in line 4 in *new-source*
6a7	Line 7 in *new-source* has been added after line 6 in *old-source*
10d10	Line 10 in *old-source* has been deleted in *new-source*

The **Context** list box in the View Options group box of the **Source Management** sheet enables you to specify the number of lines that are displayed on either side of the changed lines when changes only are selected, if required. (By default, no lines either side of changed lines are displayed.)

Discarding Changes Made to a Checked Out Method

Use the **Undo Checkout** command from the Methods menu Delta submenu in the Class Browser or Primitive Types Browser to discard any changes that you made to the method selected in the Methods list. (To enable a user to check methods in to the JADE database, a delta must be set.)

To discard any changes (undo checkout)

1. In the Methods List of the Class Browser or Primitive Types Browser, select the checked out method whose changes you want to discard.
2. Select the **Undo Checkout** command from the Methods menu Delta submenu. A Confirm dialog advises you that the method source has changed, and prompts you to confirm that you want to discard (lose) the changes to the method.
3. Click the **OK** button to confirm that you want to undo the changes to the method.

Any changes that you may have made to the selected method are then discarded and the method is checked in.

The status line no longer states that the method is checked out, and the **Undo Checkout** command is disabled.

Extracting Changed Methods

To extract all changed methods

- Select the **Extract** command from the Methods menu.

Saving changed methods provides you with a quick means of extracting only changed methods; for example, for a backup or to pass the code to another user.

Using this command is a simpler, faster way to extract specific methods than performing a selective extract of your schema.

The common Save As dialog is then displayed, to enable you to specify the name and location of your changed methods file.

The file name defaults to the name of the current schema, with a **.scm** suffix. The file location defaults to your JADE working directory.

For details about extracting selected changed classes or methods, see "Extracting Selected Entities".

Viewing Checked Out Methods

The Browse menu contains the commands listed in the following table, to enable you to view changed methods or checked out methods in the current delta.

Command	Action	For details, see…
Changed Methods	Displays the Changed Methods dialog	Viewing Changed Methods
Checked Out Methods	Displays the Checked Out Methods dialog	Viewing Selected Checked Out Methods

For details, see the following subsections. (See also the **JadePatchControlInterface** class **getCheckedOutEntitiesForPatch** method in Volume 1 of the *JADE Encyclopaedia of Classes*, about returning checked out methods for a specified patch number and delta.)

Viewing Changed Methods

Use the Browse menu **Changed Methods** command to select and display methods that have changed within specified criteria.

You can specify the display of methods that have changed:

- From one version to another specified version
- Within a specified period

You can also specify that changed methods are displayed for methods changed:

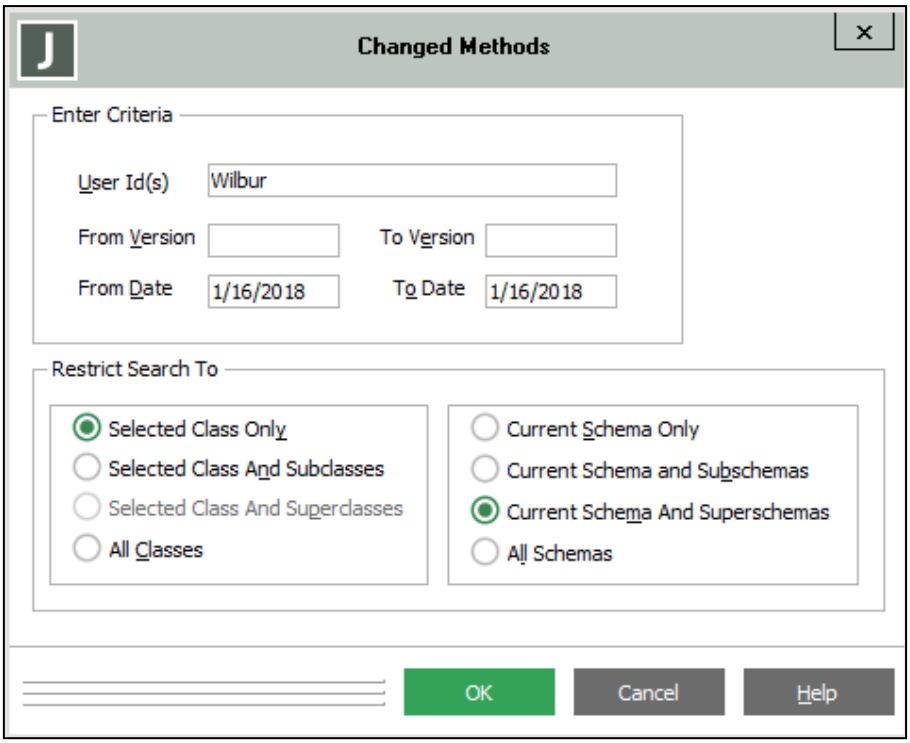
- By one or more specified users
- In the selected class and optionally its subclasses, superclasses, or in all classes
- In the selected schema and optionally its subschemas, superschemas, or in all schemas

Viewing changed methods for a class reflects any changes to the signature of interface methods that are implemented by that class. (As an interface method cannot contain a body, which is your responsibility to define in the automatically generated method in the implementing class, a separate restrictive view for interface methods is not required.)

To specify the changed methods that you want to view

- Select the **Changed Methods** command from the Browse menu.

The Changed Methods dialog, shown in the following image, is then displayed.



The Changed Methods dialog enables you to specify the changed methods that you want to view.

If you start browsing from a non-versioned schema or you select the **All Schemas** option button when browsing for changed methods in a versioned schema tree, the search results include both the current and latest contexts for any versioned schema included in the search; for example, for a search that starts with a non-versioned schema and includes superschemas.

Using the Changed Methods Dialog

Use the Changed Methods dialog to specify the changed methods that you want to view.

To select the changed methods that are displayed

1. In the **User Id(s)** text box, specify the user identifier of the user whose changed methods you want to view. If you do not enter a value in this text box, the changed methods for all users are displayed.

   You can specify one or more user identifiers in this text box, by entering each of the required identifiers separated by a space.

2. In the **From Version** text box, specify the number of the JADE version from which the changed methods are to be displayed; for example, **7.1.06**. If you do not enter a value in this text box, the search for changed methods starts from your first JADE version.

   An error is raised if you do not specify a value in this text box but you specify a value in the **To Version** text box.

3. In the **To Version** text box, specify the number of the latest JADE version in which the changed methods are to be displayed; for example, **7.1.10**.

   If you do not enter a value in this text box, the value defaults to that specified in the **From Version** text box, if present. If you specify a value in neither the **From Version** nor the **To Version** text box, the search is performed on all changed methods that match any specified user id or date criteria.

4. In the **From Date** text box, specify the starting date from which changed methods are to be displayed; for example, **03FEB2015**. Enter the date in the *ddMMMyyyy* format.

   If you do not specify a value in this text box, the search for changed methods starts at the earliest timestamp; that is, the first date on which a method was changed. An error is raised if you do not specify a value in this text box but you specify a value in the **To Date** text box.

5. In the **To Date** text box, specify the date up to which changed methods are to be displayed; for example, **30MAY2016**. Enter the date in the *ddMMMyyyy* format.

   If you do not enter a value in this text box, the value defaults to that specified in the **From Date** text box, if present. If you specify a value in neither the **From Date** nor the **To Date** text box, the search is performed on all changed methods that match any specified user id or version criteria.

6. Use the options in the Restrict Search To group box to select the classes and schemas to which you want to restrict the search for changed methods if you do not want to display only the changed methods of the current class in the current schema and its superschemas.

   Note The search is context-dependent. For example, if the selected schema has no subschemas, the **Current Schema And Subschemas** option button is disabled.

   When the Class Browser or Primitive Types Browser has focus, the search is performed only on the selected class in the current schema and its superschemas by default (that is, the **Selected Class Only** option button and the **Current Schema And Superschemas** option button are selected).

   Conversely, when the Schema Browser has focus, options that are not valid are disabled and the **All Classes** and **Current Schema And Superschemas** option buttons are selected by default.

If you want to restrict the search for changed methods to another option, the values that are you can select are listed in the following table, to enable you to select the appropriate option.

Class Restriction Options	Schema Restriction Options
Selected Class Only (Class or Primitive Types Browser default value)	Current Schema Only
Selected Class And Subclasses	Current Schema and Subschemas
Selected Class And Superclasses	Current Schema and Superschemas (default value)
All Classes (Schema Browser default value)	All Schemas

7. Click the **OK** button to initiate the search. Alternatively, you can click the **Cancel** button to abandon your selections.

Note If you do not specify any values in the Enter Criteria group box, the search is performed for all changed methods that meet your search restriction option selected in the Restrict Search To group box.

The search is then initiated, and a progress dialog displays the progress of the search for changed methods.

When the search is complete, the Change Identification Browser is then displayed.

Using the Change Identification Browser

When you have selected your changed methods search criteria in the Changed Methods dialog and clicked the **OK** button, the Change Identification Browser is displayed when the search has completed.

Use the Change Identification Browser to view all methods that have changed within your specified criteria. The method list displays in alphabetical order by class all changed methods that match your specified search criteria.

By default, user-defined methods are displayed in black and inherited methods displayed in blue.

Note Only one Change Identification Browser for the current schema can be open at any time. If a Change Identification Browser is already open for that schema, it is brought to the top when you select the **Changed Methods** command from the Browse menu. You can have concurrent open Change Identification Browsers for different schemas in a development session, if required.

To view the logic for a method

- Select the required method in the method list.

The logic for the selected method is then displayed in the editor pane.

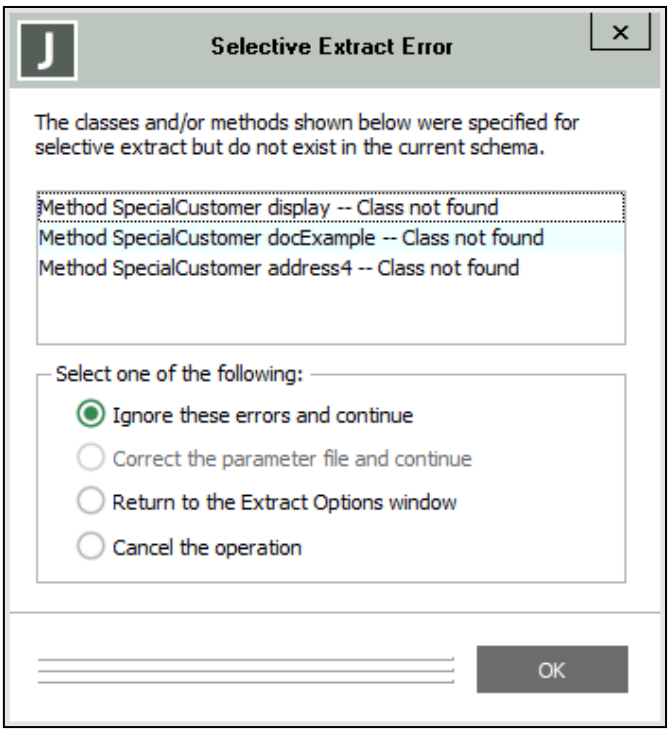
The status line displays the access mode, user id, version, and timestamp of the most recent modification for that method; for example:

```
Access is public. Modified by wilbur (7.1.10) on 25 May 2016, 08:08:24 (Checked
out)
```

You can modify and compile user-defined JADE methods.

Extracting Selected Entities

If you selected the **Selected Class And Subclasses** class restriction option in step 6 under "Using the Changed Methods Dialog" and you select the **Extract All Schema-Defined** command from the Methods menu in the Change Identification Browser or the Checked Out Methods Browser, the Selective Extract Error dialog, shown in the following image, is displayed when you click the **Save** button on the common Save As dialog if a class or method was specified for selective extract and it does not exist in the current schema (for example, you selected the **All Schemas** in the restrictions group box on the Changed Methods or Checked Out Methods dialog).



For details about extracting all schema-defined methods, see "Extracting All Schema-Defined Methods", in Chapter 10 of the *JADE Development Environment User's Guide*.

All methods or classes selected for extract but that do not exist in the current schema are displayed in the list box in the upper area of the dialog.

1. In the Select one of the following group box, if you want to ignore these errors and continue with the extract process, select the:

   - **Ignore these errors and continue** option button (the default), to extract the changed methods from the schema from which the command was selected.
   - **Correct the parameter file and continue** option button if you are using a parameter file and you want to correct the parameter file (that is, remove those entities that caused an extract error because they do not exist in the current schema) and continue with the extract process, using the replacement parameter file from which entities in error were removed.
   - **Return to the Extract Options window** option button if you want to abandon the selective extract and return to the Change Identification or the Checked Out Methods Browser.
   - **Cancel the operation** option button if you want to abandon the selective extract operation and return to the Change Identification Browser, the Checked Out Methods Browser, or the Extract dialog.

Entities that contain errors are excluded from the extract process.

2. Click the **OK** button to perform the action that you selected and return to the Change Identification or Checked Out Methods Browser.

Viewing Selected Checked Out Methods

Use the Browse menu **Checked Out Methods** command to select and display methods that have been checked out within specified criteria. You can specify the display of methods that have been checked out:

- From one delta to another specified delta, displayed in alphanumeric order
- Within a specified period

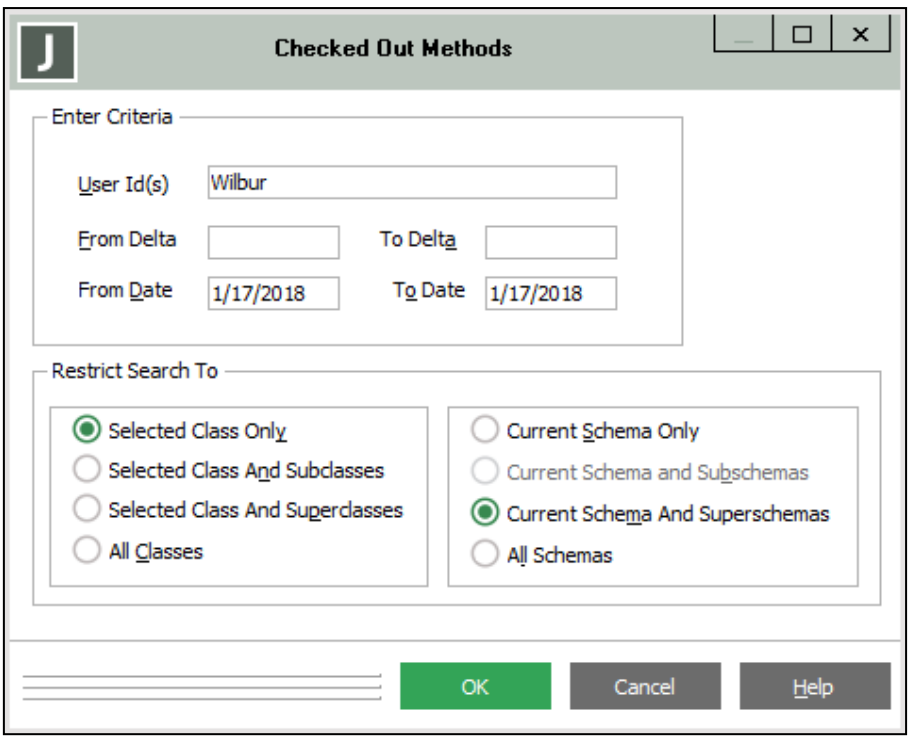
You can also specify that checked out methods are to be displayed for methods checked out:

- By one or more specified users
- Of the selected class and optionally its subclasses, superclasses, or from all classes
- Of the selected schema and optionally its subschemas, superschemas, or from all schemas

» To specify the checked out methods that you want to view

- Select the **Checked Out Methods** command from the Browse menu

The Checked Out Methods dialog, shown in the following image, is then displayed.



The Checked Out Methods dialog enables you to specify the checked out methods that you want to view.

Selecting Your Checked Out Methods for Display

Use the Checked Out Methods dialog to specify the checked out methods that you want to view.

To select the checked out methods that are displayed

1. In the **User Id(s)** text box, enter the user identifier of the user whose checked out methods you want to view. If you do not enter a value in this text box, the checked out methods for all users are displayed.

   You can enter one or more user identifiers in this text box, by entering each of the required identifiers separated by a space.

2. In the **From Delta** text box, enter the delta identifier of the delta from which the checked out methods are to be displayed; for example, **Browse**.

   If you do not enter a value in this text box, the search for checked out methods starts from your first delta.

3. In the **To Delta** text box, enter the delta identifier of the delta to which the checked out methods are to be displayed; for example, **Painter**.

   If you do not enter a value in this text box, the search for checked out methods ends at your last delta, inclusively.

4. In the **From Date** text box, enter the starting date from which checked out methods are to be displayed; for example, **03DEC2000**. Enter the date in the *ddMMMyyyy* format.

   If you do not enter a value in this text box, the search for checked out methods starts at the earliest timestamp; that is, the first date on which a method was checked out.

5. In the **To Date** text box, enter the date up to which checked out methods are to be displayed; for example, **31MAY2001**. Enter the date in the *ddMMMyyyy* format.

   If you do not enter a value in this text box, the search for checked out methods ends on the current date.

6. Use the options in the Restrict Search To group box to select the classes and schemas to which you want to restrict the search for checked out methods if you do not want to display only the checked out methods of the current class in the current schema and its superschemas.

   Note The search is context-dependent. For example, if the selected schema has no subschemas, the **Current Schema And Subschemas** option button is disabled.

   When the Class Browser or Primitive Types Browser has focus, the search is performed only on the selected class in the current schema and its superschemas by default (that is, the **Selected Class Only** option button and the **Current Schema And Superschemas** option button are selected).

   Conversely, when the Schema Browser has focus, options that are not valid are disabled and the **All Classes** and **Current Schema And Superschemas** option buttons are selected by default.

   If you want to restrict the search for checked out methods to another option, the values that are you can select are listed in the following table, to enable you to select the appropriate option.

Class Restriction Options	Schema Restriction Options
Selected Class Only (Class or Primitive Types Browser default value)	Current Schema Only
Selected Class And Subclasses	Current Schema and Subschemas
Selected Class And Superclasses	Current Schema and Superschemas (default value)
All Classes (Schema Browser default value)	All Schemas

7. Click the **OK** button to initiate the search. Alternatively, you can click the **Cancel** button to abandon your selections.

The search is then initiated, and a progress dialog displays the progress of the search for changed methods.

When the search is complete, the Checked Out Methods Browser is then displayed.

Using the Checked Out Methods Browser

When you have selected your checked out methods search criteria in the Checked Out Methods dialog and clicked the **OK** button, the Checked Out Methods Browser is displayed when the search has completed.

Use the Checked Out Methods Browser to view all methods that have been checked out within your specified criteria. The method list displays in alphabetical order by class all checked-out methods that match your specified search criteria.

Note If you selected the checking out of all references or local implementors with a method, the Checked Out Methods Browser also displays methods that match your specified check out criteria.

To view the logic for a checked out method

- Select the required method in the method list.

The logic for the selected method is then displayed in the editor pane.

If you have checked out the method, the status line displays the access, creation timestamp, that the method is checked out; for example:

```
Access is protected. Created by wilbur1 (5.2.3) on 31 July 2000, 10:17:13 (Checked
out)
```

You can modify and compile user-defined methods.

If the method is checked out by another user, a message displayed in the status line states *Cannot modify method - Is Checked Out*.

Chapter 3 Patch Versioning

This chapter covers the following topics.

Overview

Patch version control enables you to set a numeric patch version that records all additions, deletions, and updates made to schema entities (for example, methods, properties, constants, and so on) until a new patch version number is set and the patch version history is optionally removed when the changed entities in the previous patch version are extracted or a new JADE system is installed. See also "Change Control and Patch Versioning Considerations", in Chapter 1.

Patch version control covers all metadata changes; for example, interfaces, the Relational Population Service (RPS), imported WSDL files, and so on.

Patch versioning, which is completely independent of deltas and the associated change control, enables you to:

- Set up a patch number for a selected schema.
- Enable (or disable) patch version control for the JADE development database; that is, globally.
- Extract changed entities from a specified patch for a specific schema.
- Recreate a history of changes made within patch versions.
- Display a summary of all changes to a specified patch version, and optionally:
  - Exclusion of duplicate changes.
  - All changes, or those within a specified period.
  - A specific developer whose changes are to be displayed.
  - Specific entity; for example, a method in a class or a global constant.
  - Metaschema type; for example, methods, properties, or classes.
  - Additions, updates, or deletions.

  You can also compare all source changes within the patch version, and specify the order in which each element is displayed (for example, in alphabetical order of user or by type of change).
- Remove patch history changes to meet specified criteria.
- Work on different patches in the same environment.
- Set up a hook to the development security module that allows conflict resolution (where a patch is being applied to an entity that currently has an open patch number) by an external system.
- Set up development security hooks for rename actions and the JADE Painter.
- Maintain patch numbers and extract patch number sources.
- Reassign patch numbers.
- Remove a patch number history.
- Enter a patch number when signing on to the JADE development environment.

From the JADE development environment, the Browse menu provides the **Patches** command, which enables you to maintain a patch number or obtain a patch summary.

For details, see "Maintaining Patch Numbers" or "Displaying a Patch History Summary". For details about setting up patch control information for a new schema, see "Defining a Schema", in Chapter 3 of the *JADE Development Environment User's Guide*.

The JADE Installation Preferences dialog provides the Patch menu, which contains the patch version administration commands listed in the following table.

Command	For details, see…	Description
Enable / Disable	Enabling or Disabling Patch Versioning	Enables or disables patch versioning for your database

Command	For details, see…	Description
Set Patch Number	Setting Up a Patch Number	Sets a new patch version number
Extract Patch	Extracting a Patch Version	Extracts all changed entities in the current patch version
Recreate History	Recreating a History of Patch Version Changes	Recreates the history of changes in the current patch version
Remove Patch History	Removing a Patch History of Changes	Removes the patch history that matched specified criteria

For details about enabling patch control extensions (for example, to enable users to select the patch number with which they want to work), see the **EnablePatchControlExtensions** parameter under "JADE Patch Control Extensions [JadePatchControlExtensions]", in the *JADE Initialization File Reference*. When patch control extensions are enabled, the patch control hook is called for every entity that is added to the patch history, so that partial packages can be extracted and loaded.

Tip As a patch version has two keys (the JADE release version and the patch version), when you change JADE release versions the patch version is changed from 6.3.12.*patch-number* to 6.3.14.*patch-number*, for example.

If you require patch versioning when you have upgraded to a new JADE release version, you must enable patch versioning manually, as the new patch version is not automatically enabled.

If you want to override existing patch versions in entities when loading a schema (that is, you do *not* want to retain the patch number or numbers specified in a loaded schema file), see "Loading Schemas using the Schema Load Utility" or "overridePatchVersion", in the *JADE Schema Load User's Guide*, or "Specifying Advanced Load Options", in Chapter 10 of the *JADE Development Environment User's Guide*.

Patch Control for RPS Mappings

When patch control is set, changes in an RPS mapping are differentiated at the class map level. Viewing the summary of patches shows the class map that was modified in the RPS mapping. When the patch is extracted, only the modified class maps are extracted.

Patch Numbers when Loading the Forms Definition File

When loading full or partial RPS mappings in a form and data definition (.**ddb** or .**ddx**) file, patch information is created but the patch number is based on the current patch number; *not* the patch number from the load.

Patch Control Hook

The fully qualified name of the external RPS class map (**ExternalRPSClassMap**) entity in the **jadeDevelopmentPatchControl** function of the patch control hook is as follows.

```
schema-name::rps-mapping-name::class-name::table-names
```

The *table-names* value can be a comma-separated list of tables, if multiple tables are mapped using the same external RPS class map.

Changes to an RPS Mapping Not Included in a Patch Extract

Although patch control keeps track of the following types of change, they are *not* implemented in the patch extract process.

- Deleting a class map from an RPS mapping
- Changing subclass mappings within a table
- Deleting an RPS mapping

Enabling or Disabling Patch Versioning

Patch versioning is disabled by default; that is, no patch versioning is applied to any change to schemas in your JADE development database by any developer.

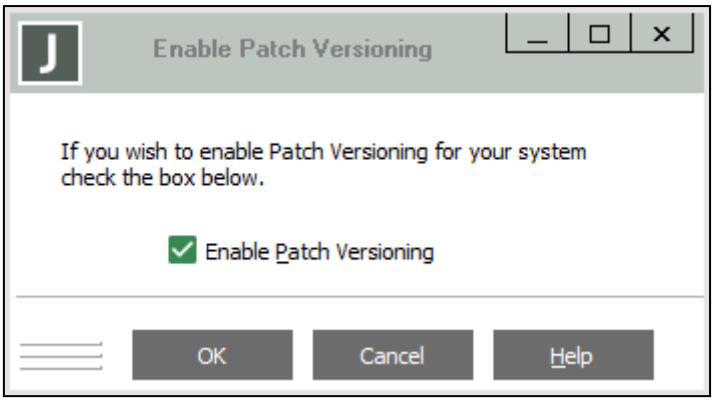
When patch extensions are enabled and a patch number is set, the current patch description and number are displayed at the upper right of the toolbar on the JADE main window.

Use the JADE Installation Preferences dialog (accessed by selecting the **Administration** option button from the Select Options group box on the JADE sign-on dialog) to enable global (system-wide) patch versioning for all changes to all schemas by all developers.

» To enable patch versioning

1. Select the **Enable** command from the JADE Installation Preferences dialog Patch menu.

   The Enable Patch Versioning dialog, shown in the following image, is then displayed.

   The **Enable Patch Versioning** check box is displayed only if you have not yet enabled patch versioning for any schema in your JADE database.

2. Check the **Enable Patch Versioning** check box to confirm that you want to enable patch versioning for the whole of your JADE database (that is, globally). You can use the patch version facilities only when you enable patch versioning by using this check box.

   For details about setting up a patch number for individual schemas, see "Setting Up a Patch Number".

3. Click the **OK** button. Alternatively, click the **Cancel** button if you do not want patch versioning applied to changes in your schemas.

When you return to the JADE development environment, patch versioning is then applied to any change made to schemas in your JADE database by any developer. The menu command is toggled to the **Disable** command.

To disable patch version control for your database when it is enabled, select the **Disable** command. (The dialog is then titled Disable Patch Versioning, the prompt asks if you want to disable patch version control for your system, and the check box caption is displayed as **Disable Patch Versioning**.

Note If global patch versioning is disabled, all other Patch menu commands are disabled; that is, you cannot set patch numbers, extract a patch, or recreate a history unless patch versioning is enabled for your JADE development environment.

Setting Up a Patch Number

The JADE Installation Preferences dialog enables you to set up a new patch version number to apply to all changes to a selected schema (and optionally any subschemas that it contains) by all developers.

For details about setting:

- Up a patch number for all schemas in your database when you have not yet set up patch versioning for any schema, see "Enabling or Disabling Patch Versioning", earlier in this chapter.
- The patch number you want to use in the current schema, see "Setting a Patch Number", later in this chapter.
- A default patch number when patch control extensions are enabled, see the **DefaultPatchNumber** parameter in the [JadePatchControlExtensions] section, in the *JADE Initialization File Reference*.
- The patch number selected in the table of the Patches Browser as the current patch number, see "Setting a Patch Number", later in this chapter.

Note You cannot set a patch number for a schema unless global patch versioning is enabled for your JADE development environment.

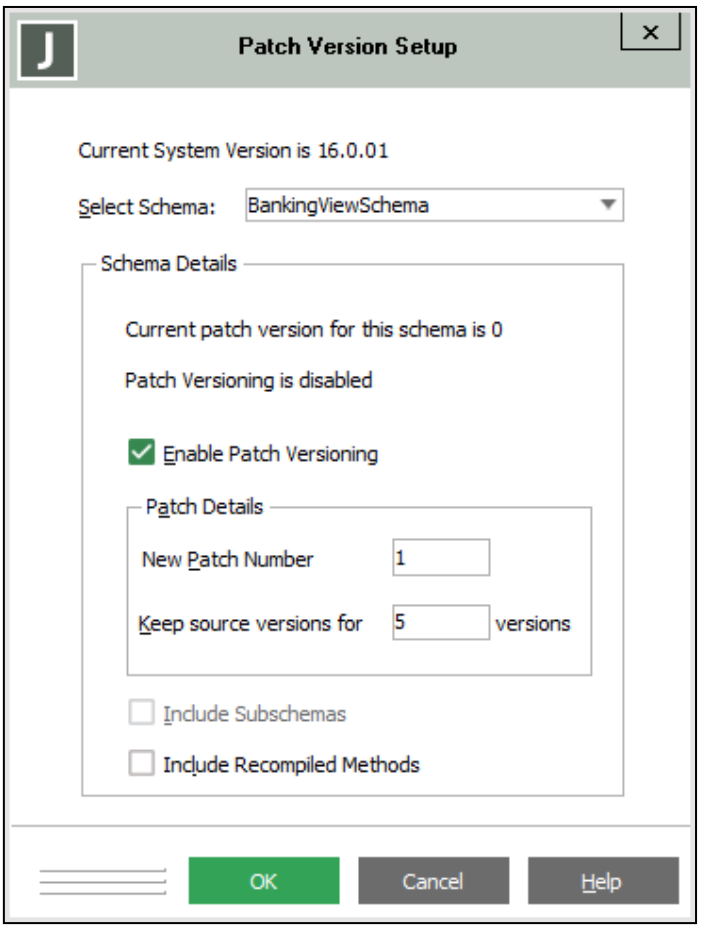
» To set up a patch version for a selected schema

- Select the **Set Patch Number** command from the JADE Installation Preferences dialog Patch menu. The Patch Version Setup dialog is then displayed.

  For details, see the following subsection, "Using the Patch Version Setup Dialog".

Using the Patch Version Setup Dialog

The Patch Version Setup dialog, shown in the following image, is displayed when you select the **Set Patch Number** command from the Patch menu in the JADE Installation Preferences dialog, to enable you to set up a new patch version for a selected schema.



The current JADE system release version is displayed at the top left of the dialog.

» To set a new patch version

1. In the **Select Schema** combo box, select the schema whose patch version you want to set up. You can select one schema only.
2. If patch versioning is currently disabled for your selected schema (indicated in the Schema Details group box), check the **Enable Patch Versioning** check box in the Schema Details group box.

   Alternatively, if patch versioning is currently disabled for your selected schema, check the **Disable Patch Versioning** check box to enable patch version control.
3. To assign a new patch version number for changes made by all developers in the selected schema, specify your new patch version number in the **New Patch Number** text box. (By default, the current number is incremented by **1**.)

If the patch number that you specify is lower than the current patch version number, a warning dialog is displayed when you click the **OK** button, advising you of this, and prompting you to click the **Yes** button if you want to continue. Click the **No** button to return to the Patch Version Setup dialog to specify a higher number, if required.

4. In the **Keep source versions** list box, select the number of changes within the current patch version number whose sources you want to keep. If the default number of five source changes is accepted and you make six changes to a method source, for example, only the latest ten changes are retained for subsequent source comparison.

   If you anticipate that schema methods may be changed several times, consider setting this to a higher value. (No warning is raised if the number of changes within the patch version exceeds the specified number, so you may later find that the first source change is no longer retained.)

5. If your selected schema has subschemas and you want these patch version options to apply also to all subschemas in that schema, check the **Include Subschemas** check box. By default, these patch version options apply only to the schema that you selected in the **Select Schema** list box; that is, you set up different patch version criteria for subschemas.

6. Check the **Include Recompiled Methods** check box if you want to include all methods that are recompiled because of other changes but whose source has not changed.

   By default, recompiled methods are not included; that is, this control is unchecked.

   Caution You should check this control when the patch is to be applied to a schema that does not have sources available. (For details, see "Removing Source Code", in Chapter 2 of the *JADE Installation and Configuration Guide*, and "Loading Schemas using the Schema Load Utility", in the *JADE Schema Load User's Guide*.)

7. Click the **OK** button. (Alternatively, click the **Cancel** button to abandon your selections.)

Patch version control is then applied to any change by developers to an entity, including schema format changes, in the selected schema (and optionally any subschemas).

Extracting a Patch Version

Use the Patch menu **Extract Patch** command from the JADE Installation Preferences dialog to extract all entities (schema formats, classes, methods, functions, properties, constants, and so on) that have patch version changes made by all developers to a selected schema. You can also specify the removal of that patch version history.

Notes You cannot extract a patch version for a schema unless global patch versioning is enabled for your JADE development environment. (For details, see "Enabling or Disabling Patch Versioning", earlier in this chapter.)

You can also remove patch history information that is older than a specified number of days, was created on or before a specified date, or to meet specific schema and patch version number criteria. (For details, see "Removing a Patch History of Changes", later in this chapter.)

When you extract all entities for a patch history, the latest version of the entity is always extracted.

As a patch extract extracts only what is defined in the patch, if a translatable string defined in an HTML document is not defined in a patch, it is not extracted with that patch.

A patch extract that has properties defined as having inverses are extracted as are the corresponding inverse definition. An inverse property that has not been updated in an extracted patch version is not extracted, and a compile error occurs when the extracted patch version is loaded into a target system in which that property is not defined.

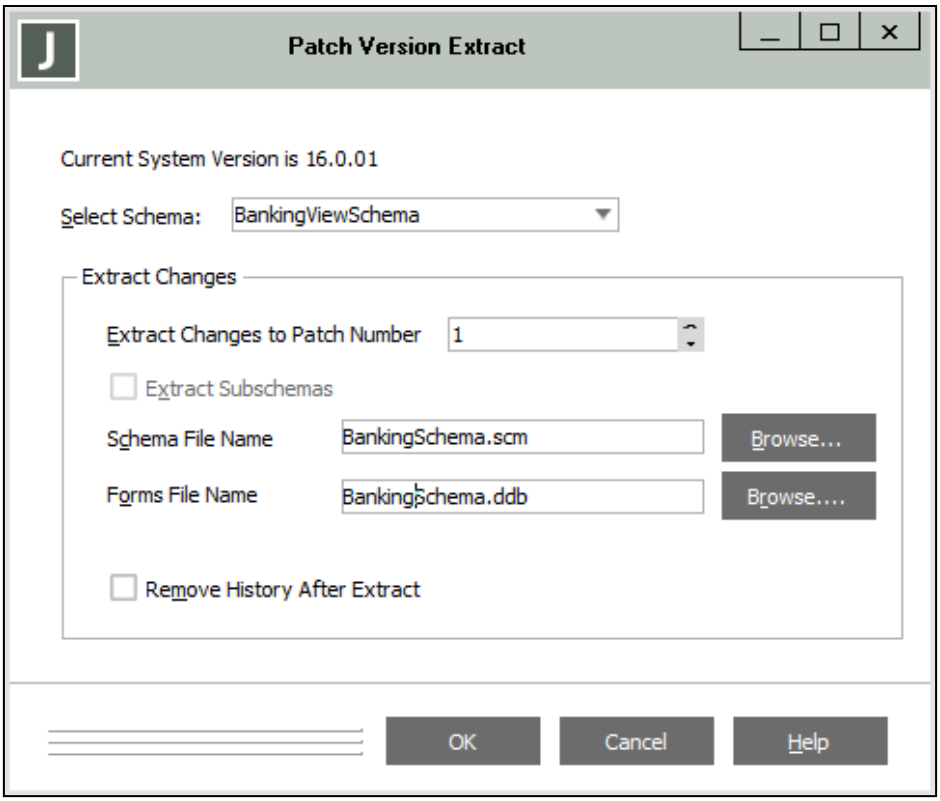
The patch version defaults to the current number. The extract process extracts only the latest version of changes. If several changes were made in the selected patch version, all changes other than the current (most recent) change is then lost when you set a new patch version number, although a summary of patch version information (that information displayed in the browser status line) is retained.

For details about using non-GUI client applications to automate the extraction of patch numbers, see "Extracting Schemas as a Non-GUI Client Application", in Chapter 10 of the *JADE Development Environment User's Guide*.

» To extract a patch version for a selected schema

- Select the **Extract Patch** command from the Patch menu in the JADE Installation Preferences dialog.

The Patch Version Extract dialog, shown in the following image, is then displayed.



For details, see "Using the Patch Version Extract Dialog", in the following subsection.

Using the Patch Version Extract Dialog

The Patch Version Extract dialog is displayed when you select the **Extract Patch** command from the Patch menu in the JADE Installation Preferences dialog, to enable you to extract patch version changes made by all developers to a selected schema. The current JADE system release version is displayed at the top left of the dialog.

For details about extracting a range of patch version changes for a specific user or selected users, see "Specifying Your Change Options", in Chapter 10 of the *JADE Development Environment User's Guide*. See also "Extracting Changes for a Patch Number", later in this chapter.

To extract a patch version

1. In the **Select Schema** combo box, select the schema whose patch version you want to extract. You can select one schema only.

2. In the **Extract Changes to Patch Number** list box, select the patch number whose changes you want to extract. You can extract changes to only one patch version at a time. The current patch number is displayed, by default. The current (latest) source change made to a method in the specified patch version is extracted.

3. If your selected schema has subschemas and you also want patch version changes extracted for subschemas, check the **Extract Subschemas** check box. By default, patch version extraction applies only to the schema that you selected in the **Select Schema** list box; that is, you extract patch version changes to each subschema separately.

   The **Multi Extract File Name** text box is then displayed, instead of the **Schema File Name** and **Forms File Name** text boxes.

4. In the **Schema File Name** text box, specify the name and location of the schema file that you want to extract; for example, **c:\jade\bin\FaultPch2.scm**. If you do not specify a location, the file is extracted to the current directory, with a default file prefix of the schema name and suffix of .**scm**.

   For ease of identification, you should prefix your schema file name with the first few characters of the schema name followed by the patch version number, and with the appropriate suffix. For example, **FaultPtch2.scm** differentiates an extracted schema not only from the **Faults.scm** file but also from changes that might have been extracted from patch version 1 as **FaultPtch.scm** or **FaultPch1.scm**.

   An error is raised if you are extracting the schema and you do not enter the name of a file or if an existing file cannot be accessed.

   If you want to extract the schema to an existing file or you are unsure of existing extract file names or location, click the **Browse** button. The common File dialog is then displayed, to enable you to select the appropriate file or location, if required.

5. In the **Forms File Name** text box, specify the name and location of the form and data definition file that you want to extract; for example, **c:\jade\bin\FaultPch2.ddb**.

   If you do not specify a location, the file is extracted to the current directory, with a default file prefix of the schema name and a suffix of .**ddb** or .**ddx**, depending on the value of the **Use DDX style (xml format) as Default instead of DDB** check box on the **Schema** sheet of the Preferences dialog.

   For ease of identification, you should prefix your forms file name with the first few characters of the schema name followed by the patch version number, and with a suffix of .**ddb** or .**ddx**. For example, **FaultPtch2.ddb** differentiates an extracted schema not only from the **Faults.ddb** file but also from changes that might have been extracted from patch version 1 as **FaultPtch.ddb** or **FaultPch1.ddb**.

   An error is raised if you are extracting the forms and you do not enter the name of a file, or if an existing file cannot be accessed.

   If you want to extract the forms to an existing file or you are unsure of existing extract file names or location, click the **Browse** button. The common File dialog is then displayed, to enable you to select the appropriate file or location, if required.

6. If you checked the **Include Subschemas** check box and you do not want the schema and subschemas patch version file extracted to the current working directory with a default file prefix of the schema name and suffix of .**mul**, specify the name and location of the patch version file that you want to extract (for example, **c:\jade\patch3\MyPatch.mul**) in the **Multi Extract File Name** text box.

   The schema and each subschema are extracted to a separate pair of .**scm** and .**ddb** or .**ddx** files. The multiple extract file itself contains merely a list of these file names.

For ease of identification if your file path does not indicate the specific patch number, you should prefix your extract file name with an identifiable name, possibly the first few characters of the schema name followed by the current patch version number.

If you want to extract the schemas and forms to an existing file or you are unsure of existing extract file names or location, click the **Browse** button. The common File dialog is then displayed, to enable you to select the appropriate file or location, if required.

7. Check the **Remove History After Extract** check box if you want to remove the patch history for your selected patch version after the source changes have been extracted.

   By default, the patch version history is *not* removed after extraction. You can therefore still view a summary of changes that were made to a previous patch version even if the changes have been overridden by subsequent changes.

   You can recreate a patch version history that you have removed, if you later want to see a summary of changes made in that patch version. For details, see "Recreating a History of Patch Version Changes", in the following section.

8. Click the **OK** button. Alternatively, click the **Cancel** button to abandon your selections.

Your selected patch version changes are then extracted and the change history for that patch removed, if applicable.

For details about extracting changes for a range of criteria (users, versions, and dates), see "Specifying Your Change Options", in Chapter 10 of the *JADE Development Environment User's Guide*.

Note Only the changes made in a schema for the specified patch version are extracted, and not the whole schema.

Extracting and Loading a Patch History

You can use the **jadclient** non-GUI client application to automate the extraction and loading of patch history, passing command line arguments after the **startAppParameters** argument to specify your extract and load requirements.

Note As patch history information is stored in compressed binary format, converting a system from ANSI to Unicode does not convert this information. This information, therefore, will not be valid when using it in a Unicode system unless you first extract the patch history from the ANSI system and load it into the Unicode system using the **Replace** option.

For details about the **jadclient** non-GUI client application, see "Running a Non-GUI Client Application using jadclient", in Chapter 1 of the *JADE Runtime Application Guide*. For details about extracting schemas using the **jadclient** program, see "Extracting Schemas as a Non-GUI Application", in Chapter 10 of the *JADE Development Environment User's Guide*.

Extracting a Patch History

To extract a patch history in a non-GUI application, specify the following arguments in the **jadclient** program.

```
jadclient path=database-path
          [ini=jade-initialization-file]
          schema=JadeSchema
          app=JadeExtractPatchHistory
          startAppParameters
          extract-arguments
```

The arguments that you can specify after the **startAppParameters** argument are listed in the following table in the order in which the values must be specified.

Argument	Description
extract-type	**A** to extract all patch numbers or **S** to extract a specific patch number.
patch-number	**0** if the extract type is **A** or the specific patch number if the extract type is **S**.
extract-path	The directory to which the patch history files are extracted. The extracted file names are the name of the schema with a file suffix of .**history**. Multiple patch numbers for the same schema reside in the same file.
extract-source	**true** if you want to extract the compressed sources or **false** if you do not want to extract sources. Note that the file sizes can be *much* smaller if you do not extract sources.
delete-after-extract	**true** if you want to delete the patch history that has been extracted or **false** if you want to retain it.
schema-name	Optional name of the schema if you want to extract patches from a single schema only.

The patch history is extracted in XML format.

Errors and messages from the **JadeExtractPatchHistory** application are written to the **jommsg** log file.

The following example extracts patch number 47474 for the **CustomerSchema** schema and places the extracted **CustomerSchema.history** file in the **d:\history** folder. It extracts sources and does not delete the patch history.

```
jadclient path=d:\jadesystems\jade\system
          schema=JadeSchema
          ini=d:\jadesystem\jade\system\jade.ini
          app=JadeExtractPatchHistory
          startAppParameters
          S 47474 d:\history true false CustomerSchema
```

Loading an Extracted Patch History

To load a patch history in a non-GUI application, specify the following arguments in the **jadclient** program.

```
jadclient path=database-path
          ini=jade-initialization-file
          schema=JadeSchema
          app=JadeLoadPatchHistory
          startAppParameters
          extract-arguments
```

The arguments that you can specify after the **startAppParameters** argument are listed in the following table in the order in which the values must be specified.

Argument	Description
load-path	The directory from which to load the patch history. All patch extract files from this directory are loaded.
abort-on-error	**true** if the operation is to be canceled if any of the files being loaded define a schema that is not in the target system or **false** if the operation is to continue.
duplicate-option	Determines the action to take when a schema and patch number combination already exists in the target system, and it must be one of the following values.

Argument	Description
	■ **A**, to cancel the operation. ■ **I**, to ignore the incoming entries. ■ **R**, to replace the existing entries. ■ **M**, to merge the incoming entries with the existing entries.

Errors and messages from the **JadeExtractPatchHistory** application are written to the **jommsg** log.

The following example loads all of the patch history files from the **d:\history** directory. It cancels the operation if any schema defined in these files is not in the target system. Where duplicates exist, they are replaced by the incoming entries.

```
jadclient path=d:\jadesystems\jade\system
          schema=JadeSchema
          ini=d:\jadesystems\jade\system\jade.ini
          app=JadeLoadPatchHistory
          startAppParameters
          d:\history true R
```

Recreating a History of Patch Version Changes

Use the Patch menu **Recreate History** command from the JADE Installation Preferences dialog to recreate a history of patch version changes (summarizing the JADE development environment status line information) following the:

- Removal of the change history during extraction, if specified.
- Migration to a new JADE system version, as your **_userscm.dat** file no longer contains your patch version changes.

  When you migrate to a new release:

  a. Extract your schemas from your current release.

     Your patch version changes are not extracted with your schema. Your source changes and patch history summary are therefore lost when you migrate to the new JADE system version. For details about extracting patch information, see "Extracting a Patch Version", earlier in this chapter.

  b. Migrate to the new JADE system version.

  c. Load your extracted schemas into the new database.

  Notes When loading schemas that contain patch information, entities that have no patch number information are loaded into the patch history set up in the target system into which the schema is loaded, by default. If you want to override existing patch versions in entities when loading a schema (that is, you do *not* want to retain the patch number or numbers specified in a loaded schema file), see "Loading Schemas using the Schema Load Utility" or "overridePatchVersion", in the *JADE Schema Load User's Guide*, or "Specifying Advanced Load Options", in Chapter 10 of the *JADE Development Environment User's Guide*.

  When loading patch information in a schema file, if the entities in the schema file have modified timestamp information, the patch history reflects this timestamp information other than the system version, which will be updated to the current system version. This is so that the history can correctly reflect the user who made the change and the date and time when the change was made.

d. Enable patch versioning. (For details, see "Enabling or Disabling Patch Versioning".)

e. Recreate a history of patch version changes, to enable you to display a patch summary.

Notes You cannot recreate a history of patch changes unless global patch versioning is enabled for your JADE development environment.

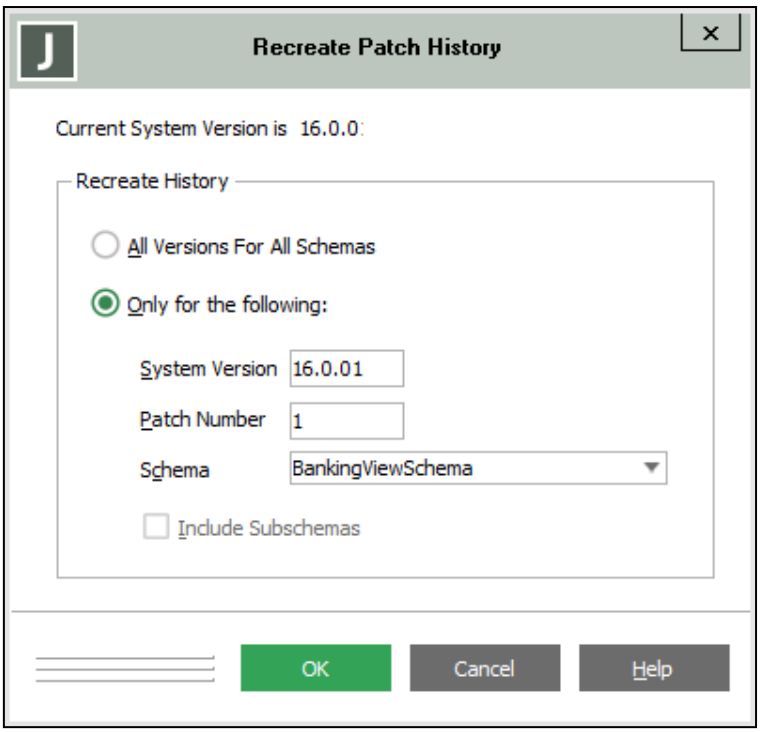
If the patch version summary is not displayed in the table in the lower portion of the Summary of Patches window, you may have checked the **Remove History After Extract** check box in the Patch Version Extract dialog or you have just migrated to or installed a new JADE system version. You must therefore recreate the patch history before you can view that summary information.

Recreating a patch version history removes versioned sources (other than the current, or latest) for the patch version.

To recreate a history of patch version changes

- Select the **Recreate History** command from the Patch menu in the JADE Installation Preferences dialog.

The Recreate Patch History dialog, shown in the following image, is then displayed.



For details, see "Using the Recreate Patch History Dialog", in the following subsection.

Using the Recreate Patch History Dialog

The Recreate Patch History dialog is displayed when you select the **Recreate History** command from the Patch menu in the JADE Installation Preferences dialog. This dialog enables you to recreate a history of changes made to a selected patch version or to all patch versions following the migration to a new JADE release or the removal of patch version history with change extraction.

Note Your patch version changes are not extracted with your schema. Your source changes and patch history summary are therefore lost when you migrate to a new JADE release version.

This **Recreate History** command enables you to recreate the patch history summary for subsequent display. Each schema entity has a timestamp, user, and patch version stored with it. Recreating a patch history simply recreates this information; for example, if a **calculateInterest** method has patch version **5.0.2.8** and you recreate the history for patch version **8**, the **calculateInterest** method is displayed in the patch summary. An entity appears once only in a recreated patch history (that is, only the most recent change is recreated).

The current JADE system release is displayed at the top of the dialog.

To recreate a patch version history

1. In the Recreate History group box, select the **All Versions For All Schemas** option button if you want patch version histories recreated for all patch versions in all schemas in your JADE system.

   By default, a patch version history is recreated only for the current patch version; that is, the **Only for the following:** option button is selected.

2. If you are recreating a patch history for a specific JADE release version and patch version (that is, the default option):

   a. In the **System Version** text box, specify the full JADE system version in which your specified patch history is to be recreated if it is not in the current system version. The current JADE system version is displayed, by default.

   b. In the **Patch Number** text box, specify the number of the patch version whose history you want to recreate. The current patch version is displayed, by default.

3. In the **Schema** combo box, select the schema whose patch history you want to recreate.

4. If your selected schema has subschemas and you want patch version histories recreated for all subschemas in that schema, check the **Include Subschemas** check box.

   By default, a patch version history is recreated only for the selected schema; that is, this check box is unchecked or it is disabled if no subschemas are defined for the selected schema.

5. Click the **OK** button. Alternatively, click the **Cancel** button to abandon your selections.

A history of patch version changes is then recreated for subsequent display. For details, see "Displaying a Patch History Summary".

Note The recreated history contains a summary of patch changes only (the information displayed in the status line of the JADE development environment); it does not recreate source changes, as these are no longer available.

Displaying a Patch History Summary

Use the Browse menu **Patches** command from the Schema Browser to display a summary of the changes made to the current schema.

Notes You cannot display a patch history summary for a schema unless global patch versioning is enabled for your JADE development environment. (For details, see "Enabling or Disabling Patch Versioning", earlier in this chapter.)

Only one Summary of Patches window for the current schema can be open at any time. If a Summary of Patches window is already open for that schema, it is brought to the top when you select the **Patches** command from the Browse menu. You can have concurrent open Summary of Patches windows for different schemas in a development session, if required.

The patches summary table uses the background color, foreground color specified in **User Objects**, and font values specified on the **Window** sheet of the Preferences dialog.

The table automatically sizes the table entries and displays a horizontal scroll bar, if required.

For details about:

- Obtaining a summary of patches, see "Maintaining Patch Numbers", later in this chapter
- Reassigning a patch history of one or more change items from one patch number to another, see "Reassigning Patch Numbers", later in this chapter

You can display the patch version summary to meet the following criteria.

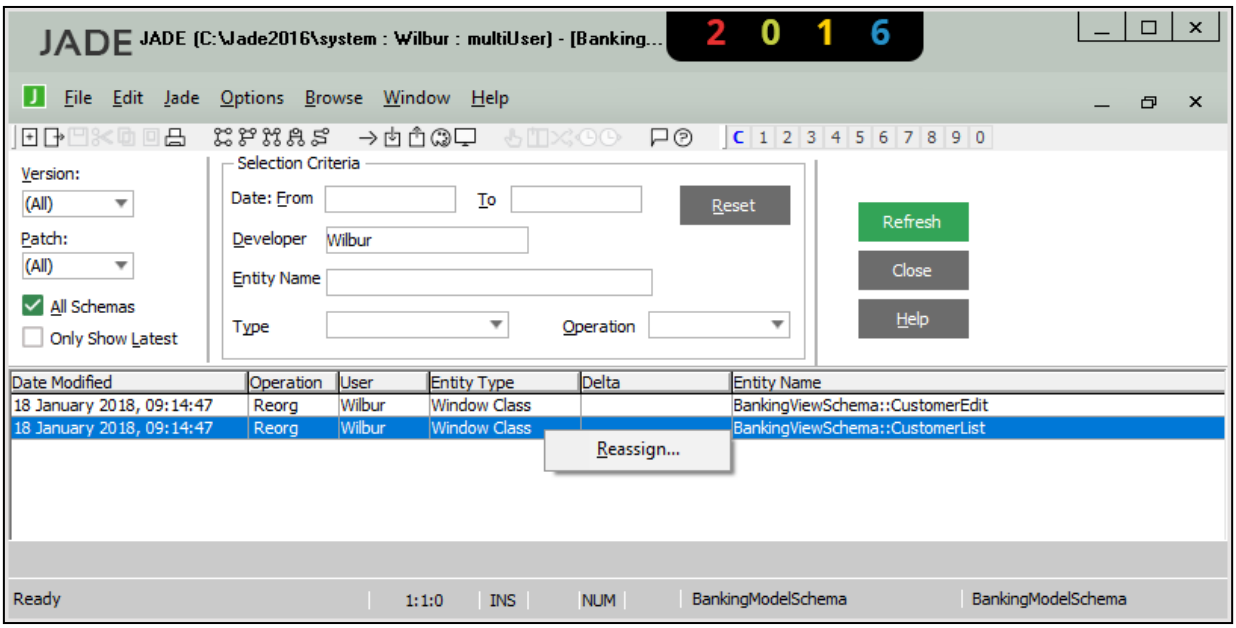
- A specific patch number.
- The current schema or all schemas
- Refine the display by specifying changes to the current schema:
  - Within specified dates.
  - By a specific developer.
  - For a specific entity.
  - For a specific type of entity (for example, a collection class).
  - By a specific type of operation (for example, all deletions only).

You can also compare all source changes within the patch version and specify the order in which each element is displayed (for example, in alphabetical order of user or by the type of change).

» To display a summary of patch version changes to the current schema

- Select the **Patches** command from the Browse menu.

When you have selected your display options and then clicked the **Refresh** button, the Summary of Patches window, shown in the following image, displays all changes that meet your selection criteria (sorted in descending order of time, by default).



The status line displays the fully qualified name of the entity selected in the summary table; that is, schema, class, and feature (method, property, or constant) name.

Using the Summary of Patches Window

Use the upper dialog area of the Summary of Patches window to specify the patch version changes that you want to view. You are then able to subsequently view them as each change is made and is output to the table in the lower portion of the window.

» To select the patch version changes that are displayed

1. In the **Version** combo box, select the JADE system version (release) whose patch version changes you want to view. If you want to display a summary of patches for all JADE releases, select the **(All)** value.

   Note If you selected the removal of the patch version history at the time that you set a new patch version or extracted a patch version, or you have just migrated to a new JADE system version, no summary information is available to display. However, you can recreate a patch history that has been removed so that you can display the most recent change to an entity in a patch. For details, see "Recreating a History of Patch Version Changes", earlier in this section.

   The current patch version is displayed, by default.

2. In the **Patch** combo box, select the patch version number whose changes you want to view. The current patch version number is selected by default.

   If you want to view a summary of changes for all open patch versions, select the **(All)** value.

3. If you want to display patch version changes in all schemas, check the **All Schemas** check box.

By default, changes are displayed only in the current schema, even though a patch number can apply across multiple schemas.

4. If you want to display only the most recent patch version change to an entity in the schema, check the **Only Show Latest** check box.

   By default, duplicate changes are displayed; for example, if two updates are made to a specific JADE method by different users, both changes are displayed.

5. In the **Date From** text box in the Selection Criteria group box, specify the starting date from which patch version changes are to be displayed; for example, **03/07/2016** or **03JULY2016**. Enter the date in the *ddMMMyyyy* format.

   If you do not specify a value in this text box, the search for changed entities starts at the earliest timestamp; that is, the first date on which an entity was changed after the specified patch version was set.

6. In the **To** text box in the Selection Criteria group box, specify the date up to which patch version changes are to be displayed; for example, **31/07/2016** or **31JULY2016**. Enter the date in the *ddMMMyyyy* format.

   If you do not specify a value in this text box, the search for changed entities ends on the current date.

7. If you want to display patch version changes for a specific developer only, specify the full user identifier (user id) of that developer in the **Developer** text box of the Selection Criteria group box.

   If you do not enter a value in this text box, the changed entities of all developers are displayed.

   You can enter only one developer user identifier in this text box. You must specify the full user id (for example, **wilburadmin**). An abbreviated user id (for example, **admin**) is not valid and would result in nothing being displayed.

8. In the **Entity Name** text box of the Selection Criteria group box, specify the name of a specific entity if you want to display patch version changes to that entity only; for example, the **ShareDemo**::**Subsidiary** class, the **JadeScript**::**testWrite** JADE method, or the **Raiders**::**Hostile** global constant.

   If you do not enter a value in this text box, all entities are displayed.

9. In the **Type** drop-down list box in the Selection Criteria group box, specify the meta schema type whose patch version changes you want to display; for example, **ExternalMethod**, **Class**, **JadeMethod**, or **PrimAttribute**.

   If you do not select a type, patch version changes to all meta schema types are displayed.

10. In the **Operation** drop-down list box in the Selection Criteria group box, select the type of operation whose changes you want to view if you do not want all additions, deletions, move actions, rename actions, reorganizations, and updates displayed.

    If you do not select a type, all operations are displayed.

11. Click the **Reset** button to clear all controls in the Selection Criteria group box; that is, specify all changes for the selected JADE system version and patch version.

12. Click the **Refresh** button to initiate the patch version display. Alternatively, click the **Close** button to abandon your selections.

The display is then initiated.

Each patch version change that satisfies any specified search criteria is then displayed in the lower portion of the window when the change is made.

To reassign a change item from one patch version to another, right-click on a change and select the **Reassign** command from the popup menu that is then displayed. For details, see "Reassigning Patch Numbers".

Your window now displays the patch version changes that meet your selected criteria as each change is made and is output to the window, in descending order of time.

Nothing is displayed if no patch history exists.

Tips To change from the default descending time sort order, click the list header for the element that you want sorted alphabetically. For example, click the **Operation** list header to list all added entities before all deleted entities and then all updated entities, or the **User** list header to specify that all changed entities are listed in alphabetical order of user id.

To display and compare method source changes, double-click the appropriate method entity in the display window.

You can resize the Summary of Patches window vertically, to enable you to view more information, if required. (You cannot resize this window horizontally, however.)

For details about specifying that you want your method source saved only when the user leaves a method rather than every time a method is compiled (in which case the previous source *and* a history entry is then saved), see "Maintaining Source Management Options", in Chapter 2 of the *JADE Development Environment User's Guide*.

Comparing a Changed Method Source

To compare the current method source with an earlier change or one change with another within the selected patch version, double-click the appropriate method entity in the Summary of Patches dialog display window.

You can change:

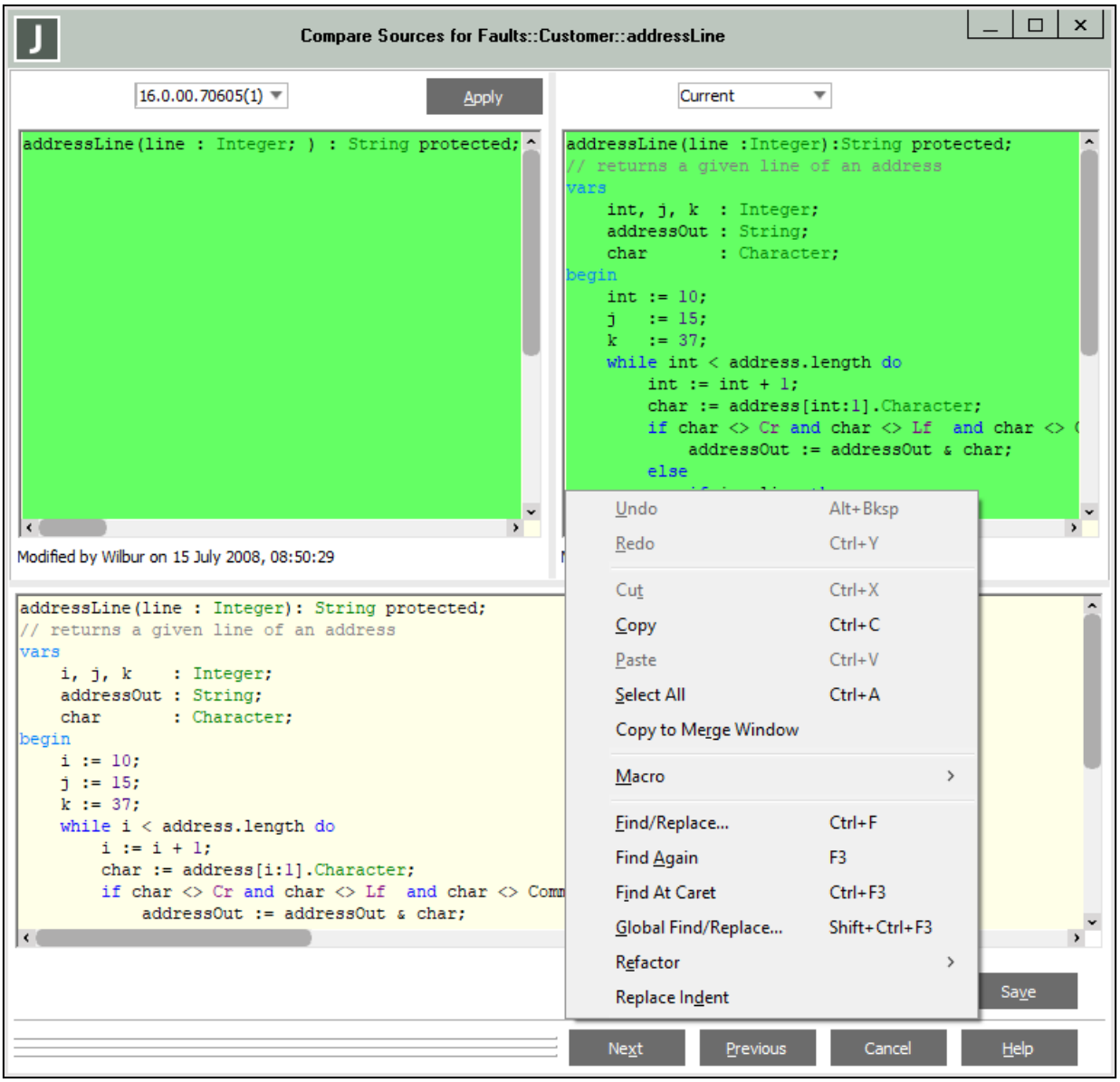
- Your default source comparison options, if required, by using the Compare Sources group box controls of the **Source Management** sheet from the Options menu **Preferences** command. For example, you can specify that white space is also to be compared.
- The size of the merge editor pane, by dragging the (splitter) resize bar.

  When the Compare Sources window is unloaded, the size of the dock control that contains the merge editor pane is also saved when the **Save Size and Position** check box is checked on the **Window** sheet of the Preferences dialog. For example, if you have dragged the splitter (resize) bar right down to get the merge editor pane out of the way and you close the form, it is not displayed when you reload the Compare Sources window until you drag up the splitter bar.

» To compare two versions of a method entity

- Double-click the method entity whose changes you want to compare.

The Compare Sources window, shown in the following image, is then displayed.



The upper part of the Compare Sources window (for the default **Visual** option in the Preferences dialog **Source Management** sheet) contains two panes, which display:

- The original source code of the method (indicated by the **1** displayed in the list box above the left-hand list)
- The current version of the source code (indicated by **Current** displayed in the list box above the right-hand list)

As these panes are read-only, you cannot change the source code in either of them.

The list boxes above both source code panes enable you to select the changes in the patch version that you want to compare. (For example, you can compare the original change with the second change, the fourth with the current change, or any combination simply by scrolling up or down both list boxes to select the changes that you want to compare.)

The merge editor pane in the lower area of the Compare Sources window enables you to copy source code from the original or updated source code pane, update it to meet your requirements, and then save it to your current delta.

If you have just migrated to a new JADE system version or extracted earlier changes and removed the patch history without having yet recreated the patch history, the pane on the left displays the following message.

```
Previous version is not available
```

If the default number of ten source changes in a patch version is accepted and you make eleven changes to a method source, for example, only the latest ten changes are displayed.

For details about increasing the number of source changes that are retained for your patches if you anticipate that your methods may be changed several times, see "Setting Up a Patch Number", earlier in this chapter.

The visual comparisons that are displayed in the Compare Sources window are listed in the following table.

Color	Description
Blue	Addition in method line
Green	Change in method line
Red	Deletion in method line

The popup menu accessed from either of the source code panes provides the **Copy to Merge Window** command, which copies all of the code in that pane to the merge editor pane. You can then change the copied source and then save it by clicking the **Save** button. The saved source is saved to your *current* delta.

The change summaries for both change versions are displayed below each list; that is, the user who modified the source for that change and the date and the time that the change was made.

To compare two versions of the selected method source

- Select the appropriate change number from the drop-down list boxes above the left and the right lists, to compare the required source changes.

To view the next change in the displayed method source

- Click the **Next** button.

The next line in the method that contains a change is then highlighted in both versions of the source.

To view the previous change in the displayed method source

- Click the **Previous** button.

The previous line in the method that contains a change is then highlighted in both versions of the source. (The **Previous** button is disabled if the first changed line is currently highlighted.)

To apply source displayed in the left of the window as the current source

1. When you decide that a version of your method source displayed in the left window should be the current (latest) version, simply click the **Apply** button above the source.

   A message box then advises you that the displayed source will become your current source if you continue.

2. Click the **Yes** button to confirm that you want to replace your existing current source with the selected source.

Your selected source is then displayed as the current source in the pane at the right of the window and your former current source in the pane at the left of the window (that is, their positions are reversed).

» To copy the entity displayed in the pane at the left of the window to the clipboard

1. Click the **Copy** button above the entity.

   A message box then prompts you to confirm that you do want to copy the contents of the left pane to the clipboard.

2. Click the **Yes** button to confirm the copy action.

You can then paste the code into a text editor or into the editor pane of another method, for example.

» To return to the Patches window when you have compared the method

- Click the **Cancel** button.

Updating and Merging Compared Source Code

You can update source code copied from the original or the updated source code pane and then save it to your current delta.

» To merge compared source code

1. Right-click in the original or the updated source code pane of the Compare Sources dialog and then select the **Copy to Merge Window** command from the popup menu that is displayed.

   All code in that source code pane is then copied to the lower merge editor pane.

2. Make any changes that you require to the source code copied to the merge editor pane.

3. Click the **Save** button.

The saved source is saved to your *current* delta.

Displaying a Patch History for a Selected Class, Type, or Method

You can display a summary of the changes made to the current class, primitive type, or method selected in the respective:

- Class List of the Class Browser
- Primitive Types List of the Primitive Types Browser
- Methods List of the Class Browser or Primitive Types Browser

Note You cannot display a patch history summary for a class, primitive type, or method unless patch versioning is enabled for your current schema or for the JADE development environment. (For details, see "Enabling or Disabling Patch Versioning", earlier in this chapter.)

» To display a summary of patch version changes to the current class, primitive type, or method

- Select the appropriate **Show History** command from the Classes, Types, or Methods menu.

The Summary of Patches window is then displayed, showing all changes to the current class, primitive type, or method in all JADE system and patch versions (sorted in descending order of time, by default). For details about using this window, see "Using the Summary of Patches Window", earlier in this chapter.

The patch history includes the method history from before the name change (until that rename entry is deleted when the number of patch entries exceeds the defined patch depth).

Notes Only name changes made after JADE 2016.0.02 show the prior history, because existing patch data does not have the required information.

Viewing the history of the new method name includes the history for the old method name but viewing the history of the previous method name does not include the history for the new name.

As you are displaying the history for the current class, primitive type, or method only, all controls in the upper dialog area other than the **Only Show Latest** check box are disabled.

If you want to view only the latest change for the current class, primitive type, or method, check this control and then click the **Refresh** button.

For details about:

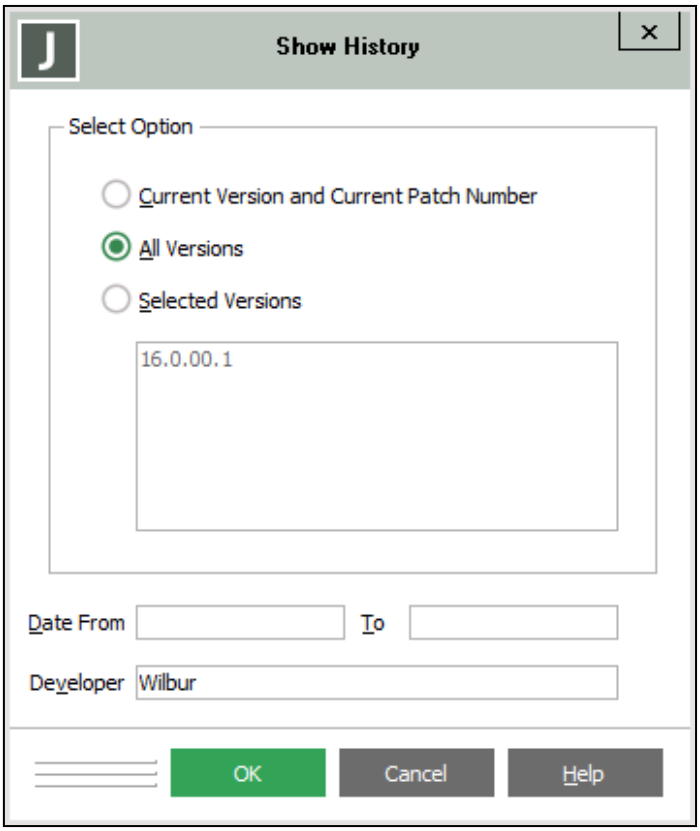
- Selecting the information that is displayed in the Summary of Patches window, see "Selecting the Patch History Information Displayed for a Class, Type, or Method", in the following subsection.
- Reassigning a patch history of one or more change items from one patch number to another, see "Reassigning Patch Numbers", later in this chapter.

Selecting the Patch History Information Displayed for a Class, Type, or Method

To select the patch history criteria that is displayed for the current class, primitive type, or method

1. Hold down the Shift key and simultaneously select the **Show History** command from the Classes, Types, or Methods menu.

The Show History dialog, shown in the following image, is then displayed.



By default, all JADE system versions and all patch number versions are displayed for the current class, primitive type, or method.

2. If you do want to display all JADE system versions and the patch number versions, perform one of the following actions.

   - Select the **Current Version and Current Patch Number** option button.
   - Select the **Selected Versions** option button.

     The JADE system and patch number versions that are applicable to the current schema are then enabled in the list box. Select one or more JADE system and patch number versions whose histories you want to view. If a JADE version has no patch number set for that version, a zero (**0**) is displayed following the JADE system version; for example, **5.2.08.0**.

3. In the **Date From** text box, specify the starting date from which JADE system and patch number version changes are to be displayed; for example, **03/07/2000** or **03JULY2000**. Enter the date in the *ddMMMyyyy* format.

   If you do not specify a value in this text box, the search for changes to the current class, primitive, or method starts at the earliest timestamp; that is, the first date on which the class, primitive, or method was changed.

4. In the **To** text box, specify the date up to which JADE system and patch number version changes are to be displayed; for example, **31/10/2000** or **31OCTOBER2000**. Enter the date in the *ddMMMyyyy* format. If you do not specify a value in this text box, the search for changes to the current class, primitive type, or method ends on the current date.

5. If you want to display JADE system and patch number version changes for a specific developer only, specify the full user identifier (user id) of that developer in the **Developer** text box of the Selection Criteria group box. If you do not enter a value in this text box, the changes to the current class, primitive type, or method by all developers are displayed.

   You can enter only one developer user identifier in this text box. You must specify the full user id (for example, **wilburadmin**). An abbreviated user id (for example, **admin**) is not valid and would result in nothing being displayed.

6. Click the **OK** button to initiate the patch history display. Alternatively, click the **Cancel** button to abandon your selections.

Each patch version change that satisfies any specified search criteria is then displayed in the table in the lower portion of the Summary of Patches window. Nothing is displayed if no patch history matching your selections exists.

For details about using this window, see "Using the Summary of Patches Window", earlier in this chapter.

Printing a Patch History Summary

» To print the current Summary of Patches window

1. Select the **Print Selected** command from the File menu. (You can print a patch history summary only when the Summary of Patches window is the current form; that is, it has focus.)

   The Print Patches dialog is then displayed, with only the option buttons in the Print Options group box enabled so that you can specify where you want the output directed.

2. If you do not want the patch history summary output to the printer (the default), select the **Print Preview** or the **RTF File** option button, as required.

3. Click the **OK** button to initiate the printing of the patch history. Alternatively, click the **Cancel** button to abandon your selection.

The patch history information that is output to your selected medium provides the following.

- Schema name.
- JADE system version (for example, **5.2.08** or **All**).
- Patch number (for example, **3** or **All**).
- Optionally selected date range.
- Optionally selected developer.
- Entity name and type.
- Operation.
- Patch summary information, as follows.
  - Data modified
  - Operation
  - Developer
  - Entity type
  - Entity name

For details, see "Printing a Selected Schema Element", in Chapter 3 of the *JADE Development Environment User's Guide*.

Removing a Patch History of Changes

Use the Patch menu **Remove Patch History** command from the JADE Installation Preferences dialog to remove a history of patch number changes (summarizing the JADE development environment status line information) that meets one of the following criteria.

- Older than a specified number of days
- Created on or before a specified date
- By selected schema or schemas for selected patch number or patch numbers

  The displayed patch numbers are only those that are valid for the selected schema or schemas.

This command enables you to remove specific patch history information without having to extract individual patch numbers and select the **Remove History After Extract** option. (For details, see "Extracting a Patch Version", earlier in this chapter.) For example, if one of your JADE developers were to accidentally turn on patch versioning and it was left on while some major development work was being done on your system, the **_userscm.dat** file could increase to several hundred M bytes, of which approximately a third may be patch number information.

You can therefore use the **Remove Patch History** command to quickly strip out the patch number information that covers the period during which it was mistakenly enabled, and subsequently reclaim the database space.

To reclaim any database space, however, you must first compact your JADE database. (For details, see "Compacting Files" and "Using the Compact Files Command", in the *JADE Database Administration Guide*.)

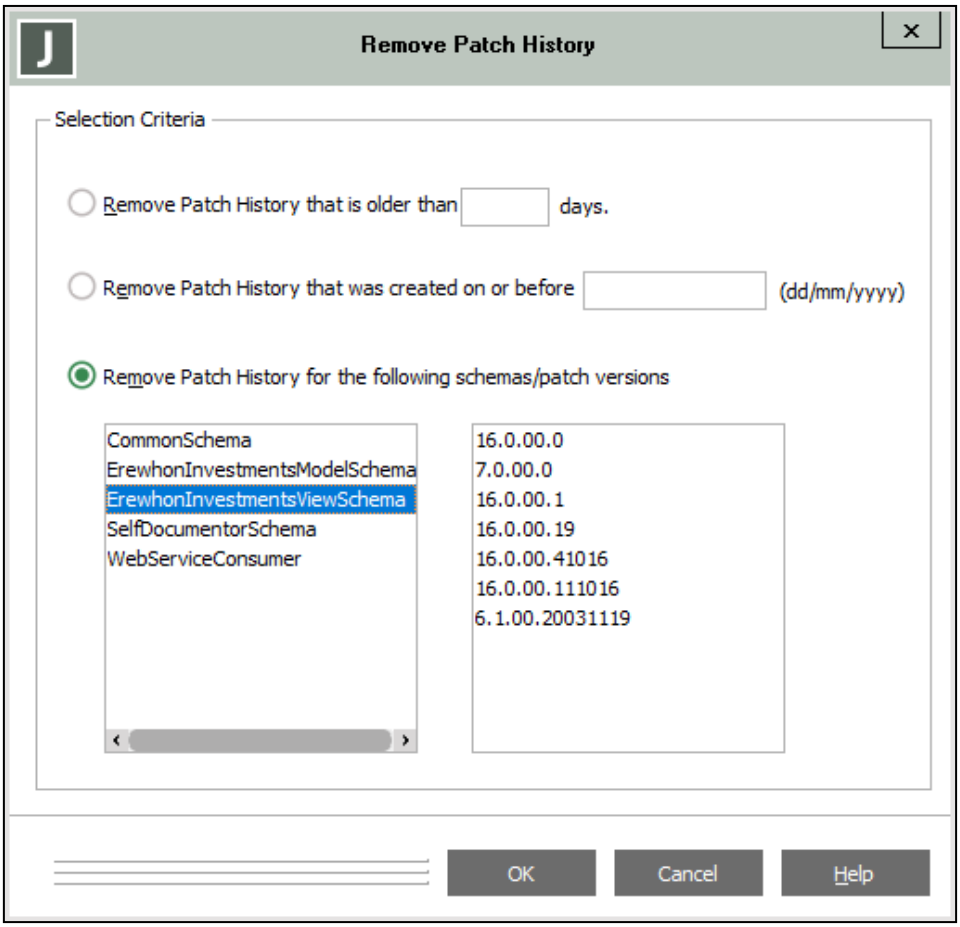
Notes You cannot remove a history of patch changes unless global patch versioning is enabled for your JADE development environment. (For details, see "Enabling or Disabling Patch Versioning", earlier in this chapter.)

You can recreate a patch version history that you have removed, if you later want to see a summary of changes made in that patch version. For details, see "Recreating a History of Patch Version Changes", earlier in this section.

To remove a history of patch version changes

- Select the **Remove Patch History** command from the Patch menu in the JADE Installation Preferences dialog.

The Remove Patch History dialog, shown in the following image, is then displayed.



Using the Remove Patch History Dialog

Use the Patch menu **Remove Patch History** command from the JADE Installation Preferences dialog to remove a history of patch version changes that is older than a specified number of days, that was created on or before a specified date, or to meet specific schema and patch version number criteria. See also "Removing a Patch History of Changes", earlier in this chapter.

To specify the patch history changes that you want to remove

1. If you want to remove patch history information that is older than a specific number of days, enter the required number of days in the text box portion of the default **Remove Patch History that is older than *nnn* days** option (where, *nnn* represents the number of days).
2. If you want to remove all patch history information created on or before a specific date:
   a. Select the **Remove Patch History that was created on or before *dd/mm/yyyy*** option button.
   b. In the text box portion of this option, specify the date prior to which all patch history information is removed, in the *dd-MM-yyyy* format (for example, **28/01/2003**).

3. If you want to remove patch history information for a specific schema or schemas and for a specific patch number or patch numbers:

   a. Select the **Remove Patch History for the following schema/patch versions** option button. The schemas and patch version numbers in the list boxes are then enabled.

   b. In the list of schemas at the left of the dialog, select the user schema or schemas whose patch history you want to remove. As the schemas list box enables multiple-selection, you can use the Shift or Ctrl key to select a group or range of schemas.

   c. In the list of patch version numbers at the right of the dialog, select the patch version number or numbers whose patch history you want to remove. As the patch version number list box enables multiple-selection, you can use the Shift or Ctrl key to select a group or range of patch version numbers, respectively.

4. Click the **OK** button to initiate the removal of the patch history that matches your selection. Alternatively, click the **Cancel** button to abandon the removal of patch history.

All patch history information that matches your selection is then removed from the JADE database.

Tip To reclaim any database space freed by the removal of patch history information, you must first compact your JADE database. (For details, see "Compacting Files" and "Using the Compact Files Command", in the *JADE Database Administration Guide*.)

Maintaining Patch Numbers

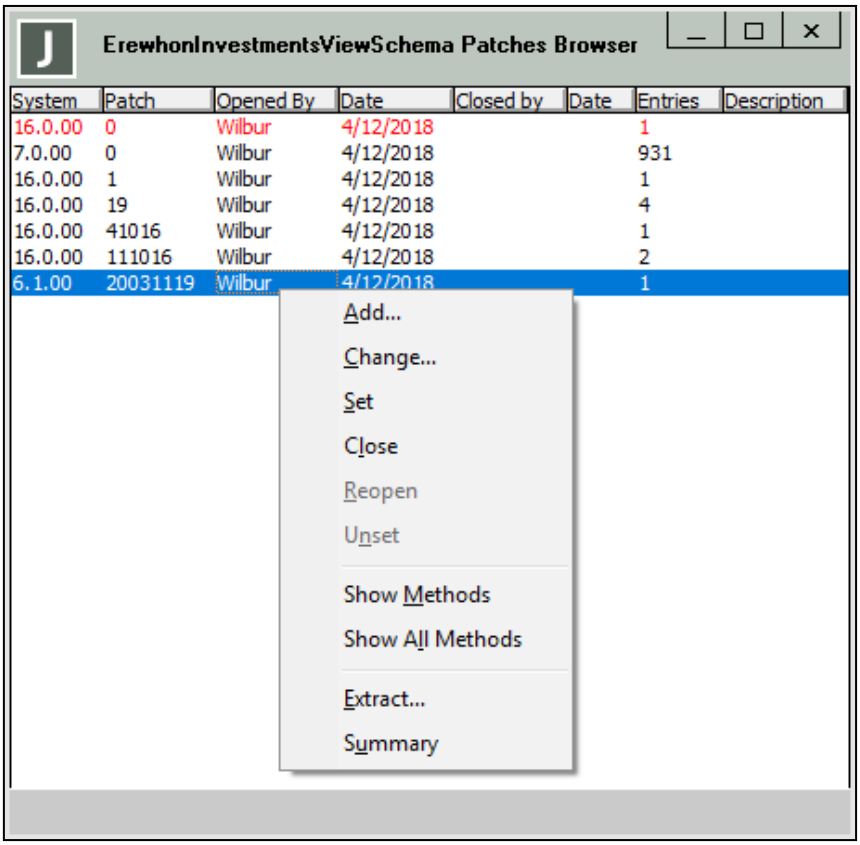
The Browser menu provides the **Patches** command, which accesses a submenu containing the **Browse** and **Summary** commands. For details about the Summary of Patches window that is displayed when you select the **Summary** command, see "Displaying a Patch History Summary", earlier in this chapter.

Note The Patches submenu is displayed only when patch control extensions are enabled; that is, when the **EnablePatchControlExtensions** parameter in the [JadePatchControlExtensions] section of the JADE initialization file is set to **true**.

» To access the Patches Browser

1. Select the **Patches** command from the Browser menu.
2. Select the **Browse** command from the submenu that is then displayed.

The Patches Browser, shown in the following image, is then displayed.



Use the Patches Browser to maintain patches and extract patches for a specified patch number. The Patches Browser displays a list of patch numbers for the current schema, as follows.

- A red entry indicates the patch number that is the currently set patch
- A black entry indicates an open patch
- A gray entry indicates a closed patch

  Closed patches are displayed only when the **All** command or the **Closed** command in the View menu is selected. By default, only open patches are displayed.

By default, all patch numbers are displayed; that is, the current patch number, all open patch numbers, and all closed patch numbers.

The Patches and View menus are displayed when the Patches Browser is the current window. For details, see the following subsections.

The Patches menu provides the commands listed in the following table. (These commands are also displayed on the context, or popup, menu when you right-click in the Patches Browser, as shown in the previous image.)

Command	For details, see…	Action
Add	Adding a Patch Number	Accesses the Patch Number Update dialog
Change	Changing a Patch Number	Accesses the Patch Number Update dialog

Command	For details, see…	Action
Set	Setting a Patch Number	Sets the current patch number to the patch number selected in Patches Browser
Close	Closing a Patch Number	Closes the selected patch number
Reopen	Reopening a Closed Patch Number	Reopens a closed patch number
Unset	Unsetting a Patch Number	Unsets the current patch number
Show Methods	Displaying Updated and Added Methods in the Current Schema	Displays the methods that were updated or added in the selected patch for the currently selected schema
Show All Methods	Displaying Updated and Added Methods for All Schemas	Displays the methods that were updated or added in the selected patch for all schemas
Extract	Extracting Changes for a Patch Number	Extracts all changes for the selected patch number
Summary	Displaying a Summary for the Selected Patch Number	Displays the Patches window for the selected patch number

The View menu provides the commands listed in the following table.

Command	For details, see…	Action
All	Displaying All Patch Numbers for the Current Schema	Displays the complete list of patches for the current schema
Open	Displaying Open Patch Numbers Only	Displays only the list of open patches
Closed	Displaying Closed Patch Numbers Only	Displays only the list of closed patches
Current User	Displaying Your Own Patch Numbers Only	Displays only the patch numbers that you opened
Refresh	Refreshing the Patches Browser	Refreshes (updates) the Patches Browser

In addition, the File menu provides the patch-related command listed in the following table.

Command	For details, see…	Action
Set Patch Number	Setting a Patch Number	Accesses the Set Patch Number dialog

For details about:

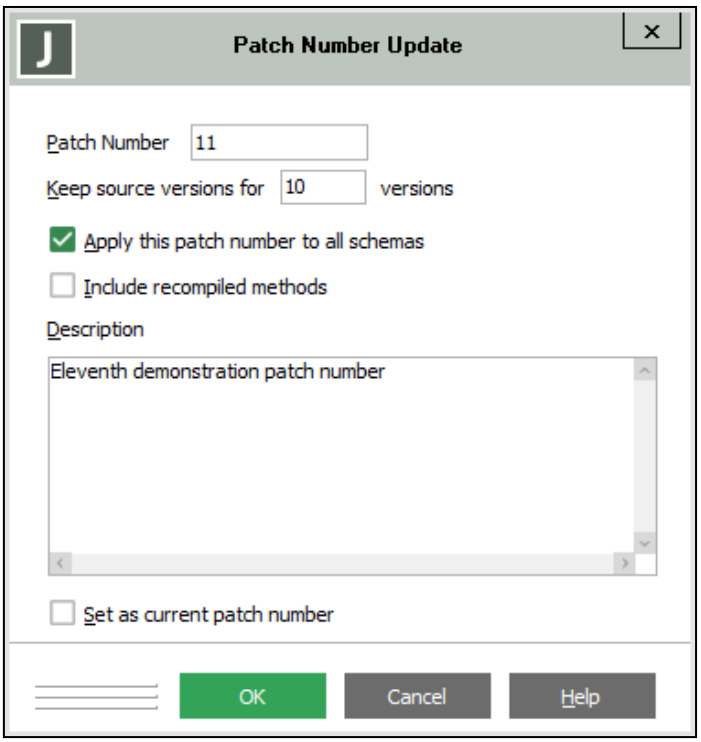
- Reassigning a patch history of one or more change items from one patch number to another, see "Reassigning Patch Numbers", later in this chapter.
- Removing a patch history, see "Removing a Patch History of Changes", earlier in this chapter.

Adding a Patch Number

To add a new patch number

1. Select the **Add** command from the Patches menu.

   The Patch Number Update dialog, shown in the following image, is then displayed.



2. In the **Patch Number** text box, specify the patch number that you require for the current schema. The patch number, which can be up to nine digits long, cannot be the same as an existing patch number in the current schema.

   Use the **AutoAssignPatchNumbers** parameter in the [JadePatchControlExtensions] section of the JADE initialization file to specify whether the patch number is automatically assigned by JADE. When the patch number is automatically assigned, the current highest patch number is incremented by 1, and cannot be changed.

3. In the **Keep source versions for** text box, specify the number of source versions that you want to retain for the current schema, with a maximum length of three digits.

   By default, 10 patch source versions are retained.

4. If you do not want to apply your specified patch number to all schemas in the JADE database, uncheck the **Apply this patch number to all schemas** check box.

   This check box, which is checked by default, is the recommended setting as it is easier to extract changes to a specified patch number if a patch applies across multiple schemas.

   To apply the specified patch number, it must be unique across all schemas.

5. Check the **Include recompiled methods** check box if you want to include all methods that are automatically

recompiled; for example, because of a signature change.

This check box is unchecked by default, indicating that recompiled methods are not included in the patch.

As a schema load cannot recompile methods that have no sources, load errors are detected when the patches are applied to a deployed system if you do not check this control.

Note Methods that are recompiled because of rename actions (for example, of a class or method) are always included in the patch, regardless of the setting of this check box.

6. In the **Description** text box, specify any free-format text that you require as descriptive text for the patch number.

   JADE automatically appends descriptive text that indicates when the patch number is opened, closed, or re-opened.

7. Check the **Set as current patch number** check box if you want to set the newly created patch number to be your current patch number.

   This check box is unchecked by default, indicating that the patch number is not set as the current patch number.

8. Click the **OK** button to add the patch number.

   Alternatively, click the **Cancel** button to abandon your selections.

Changing a Patch Number

To change an existing patch number

1. In the Patches Browser, select the patch number that you want to change.

   Note You can change only the free-format text displayed in the **Description** text box, the setting of the **Set as current patch number** check box for an open patch number, or the descriptive text of a closed patch number.

2. Select the **Change** command from the Patches menu.

   The **Change** command is disabled when you select a closed patch number.

The Patch Number Update dialog is then displayed. For details about using this dialog, see "Adding a Patch Number", in the previous subsection.

Setting a Patch Number

To set a patch number as the current patch number, perform one of the following actions

- Select the **Set Patch Number** command from the File menu in the JADE development environment.

  The Set Patch Number dialog is then displayed.

  a. If no patch number is set up and you enter a patch number in the combo box, a message box is displayed, advising you that the patch number does not exist and prompting you to click the **Yes** button if you want to create it.

     The Patch Number Update dialog is then displayed. For details, see "Adding a Patch Number", earlier in this chapter.

b. In the **Patch Number** combo box, select the patch number that you want to use in the current schema. (All recently used patch numbers in the current schema are available for selection.)

c. Click the **OK** button. Alternatively, click the **Cancel** button to abandon your selection.

The selected patch number as then set, as is indicated by it being displayed in red in the Patches Browser.

- In the Patches Browser, select the patch number that you want to set to be the current patch number and then select the **Set** command from the Patches menu. (The **Set** command is disabled when you select a closed patch or the patch number that is the current patch number.)

  The patch number is then set to be the current patch number, indicated by it being displayed in red in the Patches Browser. (For details about unsetting the current patch number, see "Unsetting a Patch Number", later in this chapter.)

The patch control hook is called with *user-name*, *patch-number*, **"", "", "", 0**. (For details, see "Patch Control Hook", in Chapter 2 of the *JADE Object Manager Guide*.) See also "Setting Up a Patch Number", earlier in this chapter.

Closing a Patch Number

» To close a patch number so that no more changes can be made in that version

1. In the Patches Browser, select the open patch number that you want to close.
2. Select the **Close** command from the Patches menu.

   The **Close** command is disabled when you select a patch number that is already closed.

The selected patch number is then closed and the user id of the person who closed the patch and the date on which it was closed are displayed in the Patches Browser for that patch number.

When you attempt to close a patch number to which other users are currently set, a message box is displayed, stating that the patch number cannot be closed and listing the other users who are set to that patch.

Reopening a Closed Patch Number

» To reopen a closed patch number

1. In the Patches Browser, select the closed patch number that you want to reopen.
2. Select the **Reopen** command from the Patches menu.

   The **Reopen** command is disabled when you select a patch number that is already open.

The patch number conflict security hook is then called for each entity in the selected patch. The hook is passed the following information.

```
(user-name, entity-patch-number:user-patch-number:status, entity-name, entity-type,
'R', 0)
```

The *entity-patch-number* value is the current patch number of the entity being modified, and is zero (**0**) if it is new or unset. The *user-patch-number* value is the patch number currently assigned to the user. The *status* value is **N** if the entity is being modified for the first time against the user's patch number, **O** if the entity has been modified previously against the user's patch number and the patch number is still open, or **C** if the entity has been modified previously against the user's patch number and the patch number is closed (for example, **0:10:N**).

If errors are returned from the security hook, the entities that are in error are displayed and the reopen action fails. (For details, see "Patch Control Hook", in Chapter 2 of the *JADE Object Manager Guide*.)

Unsetting a Patch Number

To unset the current patch number

- In the Patches Browser, select the **Set** command from the Patches menu.

The current patch number is then unset and no patch number is displayed in red in the Patches Browser. The **Unset** command is disabled when no patch is set as the current patch number. (For details about setting a patch number, see "Setting a Patch Number", earlier in this chapter.)

The patch control hook is called with *user-name*, *patch-number*, **"", "", "", 0**. (For details, see "Patch Control Hook", in Chapter 2 of the *JADE Object Manager Guide*.)

Displaying Updated and Added Methods in the Current Schema

To display the methods updated or added in the selected patch number for the current schema

- Select the **Show Methods** command from the Patches menu.

  This command is disabled when no patch is set as the current patch number or no patch is selected in the Patches Browser.

The Methods List window then displays all methods added to or changed in the current schema for the selected patch number. For details about this window, see "Using the Methods List Window", in Chapter 3 of the *JADE Development Environment User's Guide*.

Displaying Updated and Added Methods for All Schemas

To display the methods updated or added in the selected patch number for all schemas

- Select the **Show All Methods** command from the Patches menu.

  This command is disabled when no patch is set as the current patch number or no patch is selected in the Patches Browser.

The Methods List window then displays all methods added to or changed in all schemas for the selected patch number. For details about this window, see "Using the Methods List Window", in Chapter 3 of the *JADE Development Environment User's Guide*.

Extracting Changes for a Patch Number

Use the Patches menu **Extract** command to extract changes in the patch number selected in the Patches Browser. (When you select the **Changes** option on the **Extract Options** sheet of the Extract dialog, all applications added to or changed in the patch version specified on the **Changes** sheet of the Extract dialog are included in the schema extract.)

The .**jcf** command file is created from the patch history. When a full schema extract is done, command file entries are created based on all the patch histories in your JADE system. Command file entries have a comment line that states the patch number from which they were extracted.

When you extract a specific patch number, the command file contains entries relating to that patch number only.

When you perform a partial extract (by using the extract dialog), the command file contains entries for all patch numbers that are part of that extract.

In a full schema extract, the command file is necessary only when renaming or moving classes. (Renamed properties, deleted properties, and deleted classes are not required in the command file for full schema extractions.)

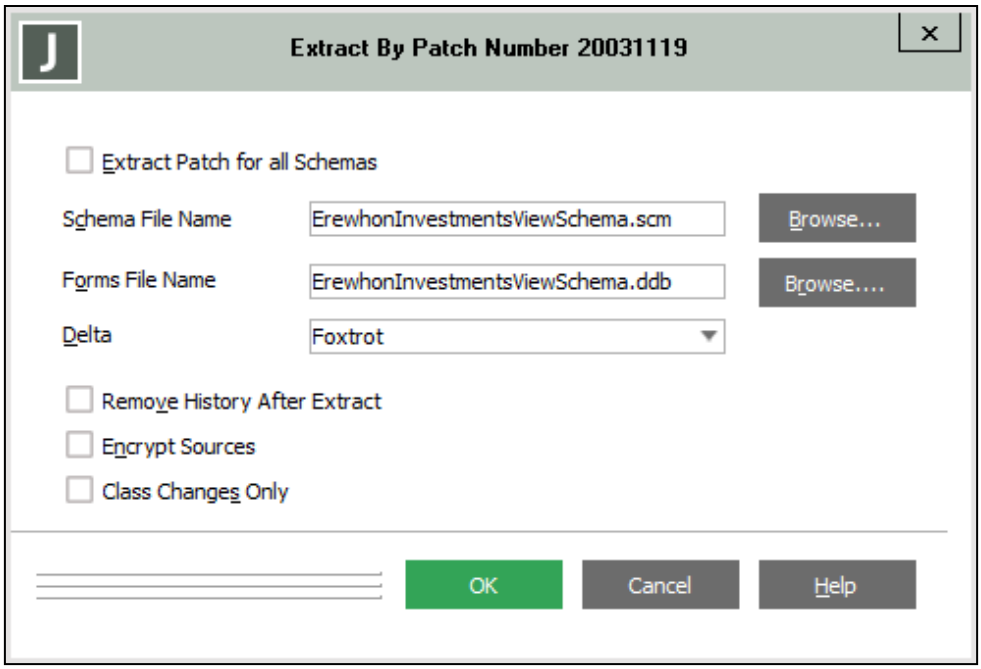
To extract all changes for a patch number

1. In the Patches Browser, select the patch number whose changes you want to extract.

   Note You can also extract closed patch numbers.

2. Select the **Extract** command from the Patches menu.

   The Extract By Patch Number dialog, shown in the following image, is then displayed.



3. If you want to extract changes made in all schemas for this patch number, check the **Extract Patch for all Schemas** check box. This check box is unchecked by default, indicating that only changes against this patch number in the selected schema are extracted.

   When you check the **Extract Patch for all Schemas** check box, the **Forms File Name** combo box is no longer displayed and the **Schema File Name** combo box is replaced by the **Multi Extract File Name** combo box, in which you specify the name that you require for your multiple schema extract file (that is, the **.mul** file). The prefix of the multiple schema extract file defaults to the name of the current schema.

   The files are extracted to the **.mul** file. The schema name is appended with **.scm** and the forms file names are the schema file name appended with **.ddb** or **.ddx**, depending on the value of the **Use DDX style (xml format) as Default instead of DDB** check box on the **Schema** sheet of the Preferences dialog. These files are created in the same directory as the multiple extract file.

4. In the **Schema File Name** text box, specify the name and location of the schema file whose changes you want to extract; for example, **c:\jade\bin\MySchema.scm**. If you do not specify a location, the file is extracted to your default directory.

If you want to extract the schema changes to an existing file or you are unsure of existing extract file names or location, click the **Browse** button. The common File dialog is then displayed, to enable you to select the appropriate file or location, if required.

5. In the **Forms File Name** text box, specify the name and location of the forms file that you want to extract; for example, **c:\jade\bin\MySchema.ddb**. If you do not specify a location, the file is extracted to your default directory.

   If you want to extract the form and control definition changes to an existing file or you are unsure of existing extract file names or location, click the **Browse** button. The common File dialog is then displayed, to enable you to select the appropriate file or location, if required.

6. In the **Delta** combo box, select the delta whose patch number changes you want to extract if you do not want to extract those checked out to the currently set delta.

   If a delta is currently set, patch number changes are extracted from that delta by default. If no delta is set, the un-checked out versions of the methods are extracted.

   For details about specifying the delta into which loaded methods are checked out, see Loading Schemas using the Schema Load Utility", in the *JADE Schema Load User's Guide*.

7. If you want to remove patch control information relating to the schema or schemas after the extraction process, check the **Remove History After Extract** check box.

   This check box is unchecked by default, indicating that patch history is not removed.

   Note To reclaim the physical space created by the removal of patch control information, you must compact the **_userscm.dat** file by using the JADE Database utility.

8. If you want to encrypt the JADE method source code in your patch number extract file, check the **Encrypt Sources** check box.

   By default, extracted source code is not encrypted; that is, this check box is unchecked. Source encryption provides security when you release schema extract files, as the source code is not easily visible (for example, when you deploy an application to a third-party). For details, see "Encrypting Schema Source Files", in Chapter 10 of the *JADE Development Environment User's Guide*.

   Caution Ensure that you extract the encrypted schema to a location different from that of your source schema. If you subsequently load the encrypted schema (for example, for testing purposes), your method source code is lost if you load it into the same database that contains your original source files, as they are not saved during the decryption process.

9. If you want to extract changes to classes only, check the **Class Changes Only** check box.

   By default, when a class definition is changed by deleting a method, property, or constant, the complete class is extracted. Check this control if you want to ignore the deletions and extract the complete class.

   This check box is unchecked by default, indicating that changes to all schema elements are extracted (for example, changes to methods, external functions, properties, and so on).

10. Click the **OK** button. Alternatively, click the **Cancel** button to abandon your selections.

For details about using non-GUI client applications to automate the extraction of patch numbers, see "Extracting Schemas as a Non-GUI Client Application", in Chapter 10 of the *JADE Development Environment User's Guide*.

Displaying a Summary for the Selected Patch Number

To display a summary of changes to a patch number

1. In the Patches Browser, select the patch number whose summary you want to display.
2. Select the **Summary** command from the Patches menu.

The Summary of Patches window for that patch number is then displayed. For details, see "Displaying a Patch History Summary", earlier in this chapter.

Displaying All Patch Numbers for the Current Schema

To display all patch numbers when only selected patch numbers are displayed

- In the Patches Browser, select the **All** command from the View menu.

All patch numbers for the current schema are then displayed and the check mark symbol is displayed to the left of the **All** command in the View menu, indicating that all patch numbers are displayed.

The patch numbers are displayed in the Patches Browser as follows.

- A red entry indicates the patch number that is the currently set patch
- A black entry indicates an open patch
- A gray entry indicates a closed patch

Displaying Open Patch Numbers Only

To display only open patch numbers for the current schema

- In the Patches Browser, select the **Open** command from the View menu.

All open patch numbers for the current schema are then displayed in black and the check mark symbol is displayed to the left of the **Open** command in the View menu, indicating that only open patch numbers are displayed.

Displaying Closed Patch Numbers Only

To display only closed patch numbers for the current schema

- In the Patches Browser, select the **Closed** command from the View menu.

All closed patch numbers for the current schema are then displayed in gray and the check mark symbol is displayed to the left of the **Closed** command in the View menu, indicating that only closed patch numbers are displayed.

For details about reopening a closed patch number, see "Reopening a Closed Patch Number", earlier in this chapter.

Displaying Your Own Patch Numbers Only

To toggle the display of your own patches only

- In the Patches Browser, select the **Current User** command from the View menu.

If the Patches Browser was displaying patch numbers opened by all developers, it is then refreshed to show only the patch numbers that you opened and the check mark symbol is displayed to the left of the **Current User** command in the View menu, indicating that only your own patch numbers are displayed.

Alternatively, if only the patch numbers that you opened were displayed, the Patches browser is refreshed so that the patch numbers opened by all developers are displayed.

Refreshing the Patches Browser

» To refresh the display of patch numbers in the Patches Browser

- In the Patches Browser, select the **Refresh** command from the View menu.

The Patches Browser is then refreshed (updated).

Reassigning Patch Numbers

The Summary of Patches window enables you to reassign a patch history of one or more change items from one patch number to another.

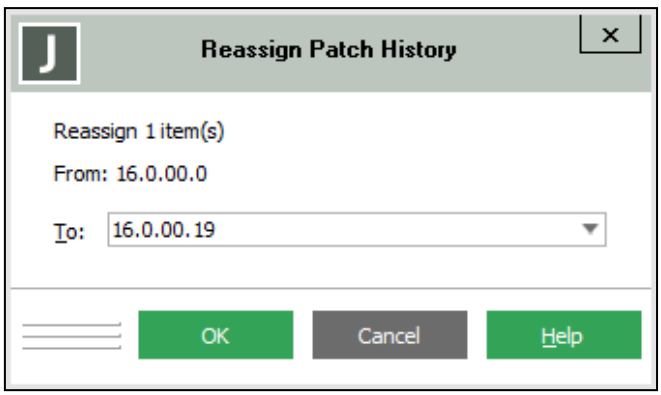
» To reassign a patch history

1. In the table in the lower area of the Summary of Patches window, right-click on the change that you want to reassign to another patch number.

   Tip If you want to reassign a group or range of change items, use the Shift key or the Ctrl key to make multiple selections.

2. Select the **Reassign** command from the pop-up menu that is then displayed.

   The Reassign Patch History dialog, shown in the following image, is then displayed.



   The **Reassign** caption may not necessarily show the number of items you selected. It usually shows more than the number of items you selected because an entry in the Summary of Patches summary window usually has more than one history entry associated with it.

3. In the **To** combo box, specify or select the patch number to which the selected change item (or items) is reassigned. The patch control hook, which has the following syntax, is called for each entity that is to be moved.

   ```
   entity-name, entity-type, 'M', entity-patch-number: user-patch-number:status
   ```

The *entity-patch-number* value is the current patch number of the entity being modified, and is zero (**0**) if it is new or unset. The *user-patch-number* value is the patch number that is currently assigned to the user.

The *status* value is **N** if the entity is being modified for the first time against the user's patch number, **O** if the entity has been modified previously against the user's patch number and the patch number is still open, or **C** if the entity has been modified previously against the users' patch number and the patch number is closed.

4. Click the **OK** button. Alternatively, click the **Cancel** button to abandon the reassignment action.

For details about the Summary of Patches window, see "Displaying a Patch History Summary", earlier in this chapter.

Reassigning Patch Numbers from a Command Script

You can reassign open patch numbers from a command script (for example, after loading schema patches in a deployed database, to reassign all open patch numbers to the latest JADE release), by specifying the following.

```
jadloadb path=database-path
         ini=initialization-file-name
         server=singleUser
         executeSchema=RootSchema
         executeClass=Schema
         executeMethod=_reassignPatchNumbers
```

Alternatively, if you want to reassign all patch numbers (that is, both open and closed path numbers), by specify the following.

```
jadloadb path=database-path
         ini=initialization-file-name
         server=singleUser
         executeSchema=RootSchema
         executeClass=Schema
         executeMethod=_reassignAllPatchNumbers
```

Note If a patch number that is being reassigned already exists in the target JADE release with a version number for that release (for example, release 6.3.11), the reassigned patch number will have two entries (for example, one for release 6.2.12 and one for 6.3.11).

This is likely to occur only if a schema load or changes to entities are made in the JADE development environment with the same numbers as those that existed in earlier release before the **_reassignPatchNumbers** or **_reassignAllPatchNumbers** method is executed.

For details about the batch JADE Schema Load (**jadloadb**) executable, the **JadeSchemaLoader** application in **jadclient**, **jade**, or the **Application** class **startApplicationWithParameter** method, see "Loading Schemas in Batch Mode using jadloadb" in the *JADE Schema Load User's Guide*.