IBM VisualAge® for Java™, Version 3.5

External Version Control

Note!
Before using this information and the product it supports, be sure to read the general information under Notices.

Edition Notice

This edition applies to Version 3.5 of IBM VisualAge for Java and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Interface to external version control systems (Windows)

The VisualAge for Java IDE offers a version control interface for checking files in and out of an external software configuration management (SCM) system. An SCM system allows developers to manage their source code. Such systems usually include features like versioning and exclusive locks for editing.

The interface to external version control systems is an optional feature that you can install when you install VisualAge for Java. This interface does not disable or interfere with the built-in software management features of the IDE.

The interface from VisualAge for Java to external version control systems uses Microsoft's Source Code Control (SCC) API. It supports the following SCM systems:

- ClearCase for Windows NT®, from Rational Software Corporation
- PVCS Version Manager, from MERANT (formerly INTERSOLV)
- VisualAge TeamConnection™ from IBM Corporation
- Visual SourceSafe from Microsoft®

For a list of the supported versions of these SCM systems, refer to the release notes.

The interface is supported for Windows NT, Windows® 98, Window 2000, and Windows clients. It may work with other SCC-compliant SCM providers, but IBM has only tested the products and releases listed above. Refer to the release notes for a list of any other supported SCM systems.

Each project in VisualAge for Java can be can be associated with an SCM system, and a specific group of files in the SCM system's source code. When you associate a VisualAge for Java project with an SCM file group, you have "added a project to version control". Each project that you add to version control can be associated with a different SCM system. You cannot, however, associate one project with more than one system. As well, you can only perform one version control operation at a time.

When you add a project to version control you have to specify which *handler* you want to use. There is one handler that is specific to IBM TeamConnection, and a generic handler for any SCM system that supports Microsoft's SCCI specification.

Once you have added a project to version control, you can add classes to your SCM systems, check classes in and out of the SCM system, and import the most recently checked-in version of a class from the SCM system using the IDE.

When you add a project to version control from VisualAge for Java, you may need to provide specific communication information in order to connect to your SCM system. For example, to use TeamConnection, you must provide information about the release and component that you want to connect to.

Adding projects to version control provides you with a convenient way to use an existing SCM system without leaving the IDE, but you will need to correlate the functions of the two systems. The version control interface does not provide any

automatic synchronization of version names between the VisualAge for Java repository and the external SCM system. VisualAge for Java does not prevent you from changing a program element in your workspace if you have not checked it out.

You should already be familiar with your SCM system and comfortable working with it before you try connecting to it from VisualAge for Java.

RELATED CONCEPTS

Repository
Team development - Overview
External SCM systems versus team development
Comparison of SCM terms

RELATED TASKS

Preparing to use an external SCM system
Adding a project to version control
Adding files to version control
Checking files out
Checking files in
Undoing check out
Deleting files
Modifying files
Removing a project from version control
Refreshing the files in your project
Launching SCM client software
Changing version control properties

External SCM systems versus VisualAge for Java team development

VisualAge for Java, Enterprise Edition, provides a team development environment that uses a shared, object-based source code repository. This is VisualAge for Java's implementation of SCM; it provides software configuration support for development projects where multiple programmers work on the same code at the same time, and where they may need to support multiple versions. This shared repository implementation, sometimes called ENVY, is used by VisualAge Smalltalk and has become a de facto standard for team development in Smalltalk environments.

Version control and repository management are integrated into VisualAge for Java, Enterprise Edition. The shared repository offers excellent support for day-to-day team programming activities. Even so, you may wish to install external SCM support as a complementary feature for one of the following reasons:

- You already use another SCM provider as your standard for application development.
- Your build process relies on external SCM providers
- You have established practices for archiving applications on a particular enterprise server, for example for disaster recovery purposes.
- You wish to manage all of your development artifacts with a single provider, or to integrate multiple programming languages across your environment.
- VisualAge for Java, Enterprise Edition, allows programmers to work concurrently on the same class. Each programmer works in a separate, unique edition of the class, and project owners must approve changes by releasing

them. This approach encourages programmers to think in terms of objects rather than files, and it fosters team communication. Nonetheless, you may be more comfortable with a traditional file checkin/checkout approach that enforces serial development of classes.

RELATED CONCEPTS

Interface to external version control systems

Comparison of SCM terms

As you use the VisualAge for Java interface to external version control systems, the following table may help you to correlate the terms that you encounter.

VisualAge for Java interface to SCM systems	**PVCS**	**ClearCase**	**VisualAge TeamConnection**	**Visual SourceSafe**
SCM repository	archives	data repository	repository	Visual SourceSafe
project	project	combination of VOB + view	combination of family + release + component + work area; sometimes known as version context	project
check in	check in	check in	check in part	check in
check out	check out	check out	check out part	check out
undo checkout	unlock revision	undo checkout	unlock part	undo checkout
add to version control	create archive	add to source control	create part	create project
refresh	check out the tip (latest version) with no read or write lock	not applicable	extract part	get latest version
comments	change description	comments	remarks	comments

RELATED CONCEPTS

Interface to external version control systems

Chapter 2. Preparing to use an external SCM system

Before you can use the VisualAge for Java interface to an external software configuration management (SCM) system, you must meet the following prerequisites:

1. You must install the SCM system's client code on your workstation. Refer to your SCM system's documentation for information about installing the SCC interface.
2. Using the native SCM software, someone has to create an organizational structure that will be used to manage your source code. You will require this organizational information when you define your SCM connection. You should be familiar with your SCM software in order to create this structure.
3. You must test your native SCM client with the structure mentioned above. A good exercise is to export a few classes from VisualAge for Java to the file system, and then check them in using your native SCM client program. This action should have the following results:
   - Your client workstation has access to at least one working directory that is recognized by your SCM system.
   - You are familiar with the organizational structures that your SCM system uses.
   - You are familiar with settings that your SCM system requires, such as drive mappings, file mounts, or the creation of views.
4. Some SCM systems may need to be reconfigured to handle four-character (.java) file extensions.

Once you have met these prerequisites, you are ready to add your projects to version control. See the related links below for more information.

The working directory
When you define the connection to your external SCM system, you may need to specify a working directory. If you were not using the automated interface from VisualAge for Java to your external SCM system, you would have to export your classes manually and then use your SCM client software to check them in and out. For example, you might export the class HanoiApplet to a directory called workdir. Since HanoiApplet is part of the package COM.ibm.ivj.examples.hanoi, VisualAge for Java would export to a file called workdir\COM\ibm\ivj\examples\HanoiApplet.java.

This is the file that you would check in and out of your SCM system's repository. With the SCM interface provided with VisualAge for Java, you can select HanoiApplet in the IDE, and then select **Tools > External Version Control > Check In** from its pop-up menu. This will export the class to a .java file and check it in for you. If you specify workdir as the default working directory when you set your SCM connection parameters, the class is automatically exported to workdir\COM\ibm\ivj\examples\HanoiApplet.java.

As mentioned in the list above of prerequisites, the working directory must be recognized by your SCM software. Some SCM software systems require the working directory be in a specific location, for example, under a certain subdirectory where the SCM client software is installed. You will need to be familiar with how your SCM software operates in order to choose an appropriate working directory.

Starting the Remote Access server
Before you can connect to your SCM system, you must have the Remote Access server running. Follow these steps to start the server:

1. In the Workbench, select **Windows > Options**.
2. The Options window opens. Select **Remote Access to Tools API**.
3. Click **Start Remote Access to Tools API.** If the server is already running, this button will be unavailable.
4. Select the **Start Remote Access to Tools API on VisualAge startup** check box. The server will now start every time the IDE is opened.

If you do not have the Remote Access server running when you try to add a project to version control, you will receive an error message indicating this. You must have the Remote Access server running whenever you are working with a project under version control.

RELATED CONCEPTS

Interface to external version control system

RELATED TASKS

Adding a project to version control
Adding files to version control
Checking files out
Checking files in
Undo checkout
Deleting files
Modifying files
Removing a project from version control
Refreshing the files in your project
Launching SCM client software
Changing version control properties

Chapter 3. Adding and removing a project from version control

Adding a project to version control

Adding a project to version control produces the following results:

- It associates a VisualAge for Java project with an SCM file group.
- If the SCM file group contains source files, they are imported onto your workstation. Java source files are imported into the VisualAge for Java repository, and added to the workspace. External resources files are imported into the local file system.

There are four possible scenarios to consider:

- The edition of the VisualAge for Java project that is in your workspace does not contain any packages or classes yet, but the SCM file group contains files that you wish to import into the project.
- The edition of the VisualAge for Java project that is in your workspace contains one or more packages and classes, and the SCM file group also contains files.
- The edition of the VisualAge for Java project that is in your workspace contains one or more packages and classes, but the SCM file group is currently empty.
- Both the VisualAge for Java project and the SCM file group are empty. This scenario is actually invalid; you cannot add a project to to version control if the project and the SCM file group are both empty.

Important: Once you have added a project to version control, do not change the project's name; changing it may cause the tool to act unpredictably.

Common steps for connecting to your SCM system
You must follow these steps to connect to your SCM system, regardless of whether your SCM file group is full or empty.

To connect to your SCM system:

1. If you have not already done so, follow the steps for preparing to use an external SCM system. These steps, which include starting the Remote Access server, are discussed as a separate topic in this documentation.
2. Select the project that you wish to associate with an SCM file group.
3. Create an open edition of the project and all packages that it contains (Enterprise only).
4. From its pop-up menu, select **Tools > External Version Control > Add to Version Control**.
5. If you have more than one SCM system installed, you will be prompted to select the SCM handler that you want to use.
   - If you want to work with TeamConnection, select **VisualAge TeamConnection** and click **OK**.
   - If you want to work with any other SCM system, select **Microsoft SCCI** and click **OK**.
6. **(**SCCI Handler only) If you selected Microsoft SCCI in the previous window (and have more than one SCM client installed), a window containing a list of systems opens. Select the system you want to use. Select the "Show advanced

options if available" check box if you want to review or change the advanced options available with the SCM system you have selected. Not all SCM systems have advanced options.

7. Depending on the SCM system you have selected to connect to, you may be prompted to provide additional information. Refer to your SCM system documentation for more information about filling in any connection windows.

Associating an existing SCM file group with a VisualAge for Java project
Follow these steps:

1. Follow the common steps for connecting to your SCM system.
2. (TeamConnection handler only) In the TeamConnection parts window, provide the requested information. Click **OK.**
3. The Select Files to Import window opens. Select the files that you want to import. In the **Path prefix that is removed on import** field, enter any subdirectory path prefixes you want removed when the files are imported into VisualAge for Java. Click **OK**.
4. (SCCI handler only) The Import Files window opens, containing a list of files that will be imported into VisualAge for Java. Click **OK**.
5. A progress indicator tells you which files were successfully imported into your project. Click **OK** to close it.
6. A green arrow  appears next to your project to indicate it has been added to version control. Arrows also appear next to any types in your project. Arrows do not appear next to packages.
   Important: The green arrow will not appear next to your project with Version 3.0 of VisualAge for Java. This is a limitation of the Version 3.0 IDE.

The latest version of the selected files are extracted from your SCM repository and any files that are not already in your workspace are imported into VisualAge for Java. Java and .class files are imported into the project and added to the workspace. Non-Java resource files are copied to the project's project resource files directory. They can be viewed in the project browser. Even if you are not able to add any of the files you selected to the workspace, the project will still be added to version control and a green arrow will appear next to it.

If any of the files you have imported have errors, the following symbol appears next to them .

Any files that you import are not automatically checked out; you must do this separately. If you plan to modify the files, then check them in, you should check them out first. See the list of related tasks below for information on how to perform these tasks.

When you are adding your project to version control, a warning window may open, indicating that VisualAge for Java was unable to detect files on your SCM server. If this occurs, you must extract the files you want to work with from your SCM system, and place them in your working directory *before* you click **OK** or **Continue**. The name of your working directory will be listed in the window. VisualAge for Java will then import the files into the workspace from your working directory.

Note: If you select not to load the files from your working directory (you will not always have this option), your project in VisualAge for Java will remain empty. It will be under version control, however, and you can create classes in it and add them to version control.

Associating existing VisualAge for Java packages and classes with an empty SCM file group

If your VisualAge for Java project contains files, and the SCM file group you are connecting to is empty, new files containing your VisualAge for Java code will automatically be created in your SCM system.

To add a project to version control, follow these steps:

1. Follow the common steps for connecting to your SCM system (excluding step 3).
2. After you have connected to your SCM system, the Add to Version Control Smart Guide appears.
3. All the classes and interfaces in your VisualAge for Java project are listed in the window. Select the types you want to be placed under source control (into your SCM system). They will continue to exist in VisualAge for Java. Select the **Select all types** check box to put all the classes and interfaces in your SCM system.
4. Click **Next**. If you have any resource files associated with your VisualAge for Java project that you want to place under source control, select them.
   - (TeamConnection handler only) You can edit the **Text files end with** field if you want change which file endings are associated with text files.
5. Click **Next**. If you want to enter a path prefix that will appear as part of your files' *pathName* when they are created in your SCM system, deselect the check box and enter the path name. This is an optional field.
6. A progress indicator tells you which files were successfully exported to your SCM system. Click **OK**.
   - (SCCI handler only) The Add Files window will open first, containing a list of files that will be added to source control. If your SCM system supports comments, you can enter one. Click **OK**.
7. A green arrow appears next to your project to indicate it has been added to version control. Arrows  also appear next to any types in your project. Arrows do not appear next to packages.
   Important: The green arrow will not appear next to your project with Version 3.0 of VisualAge for Java. This is a limitation of the Version 3.0 IDE.

Now that you have added your project to version control, you can perform any of the following tasks:

- Check your files out of your SCM system
- Add new files to your SCM system
- Modify your SCM files in the IDE
- Delete files from your SCM system
- View the Properties window to find out more about your project
- Launch your SCM client software

See the list of related tasks below for information on performing these tasks.

RELATED CONCEPTS

Interface to external version control systems

RELATED TASKS

Preparing to use an external SCM system
Adding files to version control

Checking files out
Checking files in
Undoing check out
Deleting files
Modifying files
Removing a project from version control
Refreshing the files in your project
Launching SCM client software
Changing version control properties

Removing a project from version control

Removing a project from version control disassociates that project from your SCM system. After you have removed a project from version control, you can no longer check any files in that project into or out of your SCM system.

To remove a project from version control, follow these steps:

1. Select the project you want to remove from version control.
2. From its popup menu, select **Tools > External Version Control > Remove from Version Control**.
3. Confirm that you want to remove your project from version control.
4. A message appears indicating that your project has been removed from version control.

RELATED CONCEPTS

Interface to external version control systems

RELATED TASKS

Adding a project to version control
Adding files to version control
Deleting files
Modifying files

Chapter 4. Working with files in your SCM system

Adding files to version control

As you create new classes or interfaces inside a project that has been added to version control, they will be marked with the following symbol  . This means that the program element has been created in VisualAge for Java, but has not yet been added to your SCM system. Before you can check these program elements into or out of version control, you must add them to version control.

You can only add a file to your SCM system using VisualAge for Java if it is inside a project that has been added to version control.

You can select individual classes, interfaces, or resource files that you wish to add to your SCM system. You can also select projects or packages in the IDE, as a convenient way to add all of their contained classes and interfaces to the SCM system at once. There is no other relationship between projects and packages in VisualAge for Java, and the organizational constructs in your SCM system. You will need to correlate the names, contents, and versions of your VisualAge for Java program elements with the elements stored in your SCM system.

Adding files to version control
To add classes and interfaces to version control:

1. Select the classes and interfaces that you wish to add to your SCM system, or select the project or packages that contain the classes you wish to add.
2. From the pop-up menu, select **Tools > External Version Control > Add**.
3. Depending on the SCM software you are using, you may be prompted to confirm information necessary to connect to your SCM system.
4. (SCCI handler only) The Add Files window opens, containing a list of files that will be added to source control. If your SCM provider supports comments, you can enter one. Click **OK**.
5. The files now exist in your SCM system. They will be marked with the following symbol  .

The classes and interfaces remain in your VisualAge for Java workspace and system. When you want to change them, you should check them out of the SCM system. VisualAge for Java does not enforce SCM checkout, but your SCM system may not allow you to check in changes to program elements that you have not checked out.

Adding resource files to version control
To add resource files to version control, follow these steps:

1. Select the project you wish to work with and double-click to view it in the project browser.
2. Select the **Resources** tab. Any resource files that have not been added to version control are marked with the following symbol .
   - If you are using VisualAge for Java, Version 3.5, select the resource files that you wish to add to version control. From its pop-up menu, select **Tools > External Version Control > Add**.

- If you are using VisualAge for Java, Version 3.0x, select **Resource > Tools > External Version Control > Add** instead of selecting any resource files, as your selections may be ignored.

3. Depending on the SCM software you are using, you may be prompted to confirm information necessary to connect to your SCM system.
4. If you are working with Version 3.0x, the Select Project resources window opens. Select the files you want to add to version control. Click **OK**.
5. (SCCI handler only) The Add Files window opens, containing a list of files that will be added to source control. If your SCM provider supports comments, you can enter one. Click **OK**.
6. The files now exist in your SCM system. They will be marked with the following symbol  .

RELATED CONCEPTS

Repository
Resource files

RELATED TASKS

Checking files out
Checking files in
Undo checkout
Deleting files
Modifying files
Adding a project to version control
Removing a project from version control

Checking files out

In the IDE, you can select classes, interfaces, and resource files that you wish to check out of your external version control system. When you take this action, VisualAge for Java automatically does the following things:

- Replaces the editions that were in your workspace with new open editions from your SCM system.
- Locks the new editions in your workspace.

You can only check out classes, interfaces, or resource files that have been added to version control. See the list of topics below for links to related information.

Checking classes and interfaces out
To check classes and interfaces out, follow these steps:

1. From the Workbench window, select the classes and interfaces that you wish to check out, or select the project or packages that contain the classes you wish to check out.
2. Right-click the selected item, and select **Tools > External Version Control > Check Out** from the pop-up menu.
3. Depending on the SCM system you are using, you may be prompted to confirm information necessary to connect to your SCM system.
4. (SCCI handler only) The Check Out Files window opens, containing a list of files that will be checked out. If your SCM provider supports comments, you can enter one. Click **OK**.
5. The latest versions of your files are imported into your workspace.

6. If you have modified any of the selected files since you last imported them from the SCM server into the IDE, a window opens containing the list of files you have modified. You can select to keep the modified items, or refresh them with the SCM system version. Move any files you wish to refresh from the **Keep the modified items** list to the **Refresh** list.
7. Click **OK**. The classes and interfaces are checked out (unless you have selected to keep any modified items instead of refreshing them), and appear in the workspace with the locked symbol  beside them.

The classes and interfaces remain checked out of the SCM system until you check them in again, or undo checkout.

Checking resource files out

To check out non-Java resources, such as audio clips or HTML files, follow these steps:

1. From the Workbench, select the project you wish to work with and double-click it to view it in the project browser. Click the **Resources** tab.
   - If you are using VisualAge for Java, Version 3.5, select the resource files that you wish to check out. From its pop-up menu, select **Tools > External Version Control > Check Out**.
   - If you are using VisualAge for Java, Version 3.0x, select **Resource > Tools > External Version Control > Check Out** instead of selecting any resource files, as your selections may be ignored.
2. Depending on the SCM system you are using, you may be prompted to confirm information necessary to connect to your SCM system.
3. If you are using Version 3.0x, the Select Project Resources window opens. Select the resource files you want to check out. Click **OK**.
4. (SCCI handler only) The Check Out Files window opens, containing a list of files that will be checked out. If your SCM provider supports comments, you can enter one. Click **OK**.
5. The latest versions of your files are imported into your project's project resources directory.
6. If you have modified any of the selected files since you last imported them from the SCM server into the IDE, a window opens containing the list of files you have modified. You can select to keep the modified items, or refresh them with the SCM system version. Move any files you wish to refresh from the **Keep the modified items** list to the **Refresh** list. Click **OK**.
7. The resource files are checked out (unless you have selected to keep any modified items instead of refreshing them), and appear in the workspace with the locked symbol  beside them.

The resource files remain checked out of the SCM repository until you check them in again, or undo checkout.

RELATED CONCEPTS

Interface to external version control systems

RELATED TASKS

Adding a project to version control
Adding files to version control

Checking files in
Undo checkout
Changing version control properties

Checking files into your SCM system

If you have checked classes, interfaces or resource files out of your SCM system, you must check them back in again before anyone else can check them out. Any changes to the files you make while they are checked out are not visible to others users until the files are checked back in.

In the VisualAge for Java IDE, you can select individual classes, interfaces, or resource files that you wish to check into your SCM system. You can also select projects or packages in the IDE, as a convenient way to check all of their contained classes and interfaces into the SCM system at once. There is no other relationship between projects and packages in VisualAge for Java, and the organizational constructs (for example, project files, folders, versioned object bases, views, or work areas) in your SCM system.

Before you can check classes in, they must be in a project that has been added to version control. As well, you may not be able to check in a file if it has not been modified since it was checked out. See the list of related links below for more information.

To check classes and interfaces in, follow these steps:

1. From the Workbench window, select the classes and interfaces that you wish to check in, or select the project or packages that contain the classes you wish to check in.
2. From the pop-up menu, select **Tools > External Version Control > Check In**.
3. Depending on the SCM system you are using, you may be prompted to confirm information necessary to access it.
4. (SCCI handler only) The Check In Files window opens, containing a list of files that will be checked into your SCM system. If your SCM provider supports comments, you can enter one. Click **OK**.

If the files were locked or modified, these symbols have disappeared. They will be marked only with this symbol  .

To check in resource files, follow these steps:

1. From the Workbench, select the project you wish to work with and double-click it to view it in the project browser. Click the **Resources** tab.
   - If you are using VisualAge for Java, Version 3.5, select the resource files that you wish to check in. From the pop-up menu, select **Tools > External Version Control > Check In**.
   - If you are using VisualAge for Java, Version 3.0x, select **Resource > Tools > External Version Control > Check In** instead of selecting any resource files, as your selections may be ignored.
2. Depending on the SCM system you are using, you may be prompted to confirm information necessary to access it.
3. If you are using Version 3.0x, the Select Project Resources window opens. Select the resource files you want to check in.

4. (SCCI handler only) The Check In Files window opens, containing a list of files that will be checked into your SCM system. If your SCM provider supports comments, you can enter one. Click **OK**.

If the files were locked or modified, these symbols have disappeared. They will be marked only with this symbol  .

RELATED CONCEPTS

Interface to external version control systems

RELATED TASKS

Adding a project to version control
Adding files to version control
Checking files out
Undo checkout
Changing version control properties

Undo checkout

Normally, after you check classes or interfaces out of the SCM system, you modify them and then check them in again. If you decide not to check in your changes, you should cancel (or undo) your checkout to make the program elements available to other users of the SCM repository.

Note: Undo checkout does not check the file back in. It cancels the checkout so the file is no longer checked out or locked.

To cancel the checkout of classes and interfaces:

1. From the Workbench window, select the classes and interfaces that you wish to make available again. To select all classes within a project or package, select the project or package.
2. From the pop-up menu, select **Tools > External Version Control > Undo Checkout**.
3. Depending on the SCM system you are using, you may be prompted to confirm information necessary to connect to your SCM system.
4. (SCCI Handler only) A window opens containing a list of files for which you have selected to undo checkout. Click **OK**.

The files are now no longer checked out or locked.

To cancel the checkout of resource files such as graphics files or HTML files:

1. From the Workbench, select the project you wish to work with and double-click it to view it in the project browser. Click the **Resources** tab.
   - If you are using VisualAge for Java, Version 3.5, select the resource files that you wish to undo checkout for. From its pop-up menu, select **Tools > External Version Control > Undo Checkout**.
   - If you are using VisualAge for Java, Version 3.0x, select **Resource > Tools > External Version Control > Undo Checkout** instead of selecting any resource files, as your selections may be ignored.
2. From the pop-up menu, select **Tools > External Version Control > Undo Checkout**.
3. Depending on the SCM system you are using, you may be prompted to confirm information necessary to connect to your SCM system.

4. If you are working in Version 3.0x, the Select Project Resources window opens. Select the resource files for which you you want to undo checkout.
5. (SCCI Handler only) A window opens containing a list of files for which you have selected to undo checkout. Click **OK**.

The files are now no longer checked out or locked. Other developers can now check the classes, interfaces, and resource files out of the SCM repository. Depending on the external SCM system that you use, a backup copy of the classes and interfaces may be created on your workstation.

If you modify a file after checking it out, then undo the checkout, the modifications will still exist in VisualAge for Java.

RELATED CONCEPTS

Interface to external version control systems

RELATED TASKS

Adding a project to version control
Adding files to version control
Checking files out
Checking files in

Deleting files from version control

When you select to delete files from version control, by default, they will only be deleted from your SCM system. They will remain in your VisualAge for Java workspace, marked as new  . You can delete them from your workspace by selecting them and pressing the **Delete** key. They will still exist in the repository.

You can set an option in the Properties window so that when you delete files from version control, they are also automatically deleted from your workspace.

You cannot delete a file from your SCM system that has been checked out or locked. You can always delete a file from your workspace. Deleting a file from your workspace does not remove it from your SCM system.

In the VisualAge for Java IDE, you can select individual classes, interfaces, and resource files that you wish to delete. You can also select projects or packages in the IDE, as a convenient way to delete all of their contained classes and interfaces at once.

Deleting classes and interfaces
To delete classes or interfaces from version control, follow these steps:

1. From the Workbench window, select the classes and interfaces that you wish to delete, or select the project or packages that contain the classes you wish to delete.
2. From its pop-up menu, select **Tools > External Version Control > Delete**.
3. Depending on the SCM system you are using, you may be prompted to confirm information necessary to connect to your SCM system.
4. (SCCI handler only) The Delete Files window opens, containing a list of files that will be deleted from source control. If your SCM provider supports comments, you can enter one. Click **OK**.

5. The files you selected to delete will now be marked as new  . To delete them from your workspace, select them and press the **Delete** key.

Deleting resource files
To delete resource files from version control, follow these steps:

1. From the Workbench, select the project you wish to work with and double-click to view it in the project browser. Click the **Resources** tab.
   - If you are using VisualAge for Java, Version 3.5, and select the resource files that you wish to delete. From its pop-up menu, select **Tools > External Version Control > Delete**.
   - If you are using VisualAge for Java, Version 3.0x, select **Resource > Tools > External Version Control > Delete** instead of selecting any resource files, as your selections may be ignored.
2. Depending on the SCM system you are using, you may be prompted to confirm information necessary to connect to your SCM system.
3. If you are using Version 3.0x, the Select Project Resources window will open. Select the resource files you want to delete from version control.
4. (SCCI handler only) The Delete Files window opens, containing a list of files that will be deleted from source control. If your SCM provider supports comments, you can enter one. Click **OK**.
5. The files you selected to delete will now be marked with as new  . To delete them from your workspace, select them and press the **Delete** key.

RELATED CONCEPTS

Interface to external version control systems
Repository

RELATED TASKS

Adding a project to version control
Checking files out
Checking files in
Undo checkout
Adding files
Modifying files
Removing a project from version control
Changing version control properties

Modifying files

You can edit files that have been added to an external version control system in the same manner as you would normally modify VisualAge for Java files.

If you edit a locked file, the lock symbol becomes yellow  .

If you edit an unlocked file, after you save your changes, the file will be marked with a yellow triangle  . As well, when you save your changes, a warning message opens, indicating that the file was not checked out and may conflict with the version in your SCM system. This warning is optional, and you can select to turn it off in the Properties window.

RELATED CONCEPTS

Interface to external version control system

RELATED TASKS

Adding files to version control
Checking files out
Checking files in
Undo checkout
Deleting files
Changing version control properties

Refreshing the files in your project

Refreshing a project that has been added to version control brings the most current version of files from your SCM system into your workspace. You cannot refresh a project unless it has been added to version control.

Files will only be refreshed if their contents have changed since you created the project or since they were was last refreshed. If the contents of the files in the VisualAge for Java project already match the contents of the SCM system, the refresh is cancelled.

To refresh a project under version control, follow these steps:

1. Select the appropriate project.
2. From its pop-up menu, select **Tools > External Version Control > Refresh Project**. The Refresh Project window opens.
3. The Refresh Items page contains a list of files that exist in your SCM system, but not in the project that you are refreshing. As well, it contains a list of files that have been modified in your SCM system, but not in VisualAge for Java. Select the files you want to import into VisualAge for Java.
4. The Modified Files page contains a list of any classes, interfaces, or resource files you have locally modified. Select any of the items you wish to refresh and click **Add**. To remove any items, select them and click **Remove**.
5. The Checked Out page contains a list of files that you had previously checked out of the SCM repository, but that have changed in the SCM repository since then. (This can happen if your SCM system supports checkout without locking.) You can select to keep the version that is currently in your workspace or unlock the version on your SCM repository and import it into your workspace, replacing the version you currently have.
6. The Different Version page contains a list of files that have been modified in both VisualAge for Java, and in your SCM system. You can select to keep the versions currently in your workspace, or replace (refresh) your versions with the repository versions.
7. The Removed page contains a list of files that still exist in VisualAge for Java, but have been deleted from your SCM repository. If these files have not been modified in VisualAge for Java since you imported them, they are automatically deleted. If the files have been modified, you can select to delete them or to keep them.
8. Click **OK** when you have finished selecting which files you want to refresh.

If you were prompted to import your SCM files from your working directory when you added your project to version control, you must extract the latest copies of your files from your SCM system and place them in your working directory before you try refreshing your project. Otherwise, you will not be able to refresh your project properly.

RELATED CONCEPTS

Interface to external version control systems

RELATED TASKS

Preparing to use an external SCM system
Adding a project to version control
Adding files to version control
Checking files out
Checking files in
Undo checkout
Deleting files
Modifying files
Removing a project from version control
Launching SCM client software
Changing version control properties

Chapter 5. Launching SCM client software

VisualAge for Java provides a seamless interface from the IDE to several popular software configuration management systems. At times, you may also need to launch the native client software for your SCM system, for example, to access advanced features such as report generation or to access defect management software.

To launch your SCM client software from inside the IDE:

1. From the Workbench window, select any project that has been added to version control.
2. From the pop-up menu, select **Tools > External Version Control > Open Version Control**.

The SCM software that your project is associated with opens.

RELATED CONCEPTS

Interface to external version control systems

RELATED TASKS

Preparing to use an external SCM system
Adding a project to version control

Chapter 6. Changing version control properties

If you use the VisualAge for Java interface to external version control systems, you can change the default options that it uses. For example, you can change the connection information for your project. Or, you can select to have types and resource files automatically deleted in the IDE when you delete them from your SCM system.

To view or change version control properties:

1. In the Workbench, select the project, package, or type whose properties you want to work with.
2. From the pop-up menu, select **Tools > External Version Control > Properties**. The Properties window opens.
3. Click the various tabs in the notebook. For more information on each page, press F1.

RELATED CONCEPTS

Interface to external version control systems

RELATED TASKS

Preparing to use an external SCM system
Adding a project to version control
Adding files to version control
Checking files out
Checking files in
Undo checkout
Deleting files
Modifying files
Removing a project from version control
Refreshing the files in your project
Launching SCM client software

Notices

Note to U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Lab Director
IBM Canada Ltd.
1150 Eglinton Avenue East
Toronto, Ontario M3C 1H7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow the customer to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

- AIX
- AS/400
- DB2
- CICS
- CICS/ESA
- IBM
- IMS
- Language Environment
- MQSeries
- Network Station
- OS/2
- OS/390
- OS/400
- RS/6000
- S/390
- VisualAge
- VTAM
- WebSphere

Lotus, Lotus Notes and Domino are trademarks or registered trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli Enterprise Console and Tivoli Module Designer are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

Encina and DCE Encina Lightweight Client are trademarks of Transarc Corporation in the United States, or other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, or other countries, or both.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Intel and Pentium are trademarks of Intel Corporation in the United States, or other countries, or both.

Other company, product, and service names, which may be denoted by a double asterisk(**), may be trademarks or service marks of others.