

VisualAge for Java, Version 2.0



# External SCM Tools



---

# Contents

<b>Legal Notices</b> . . . . .	v
Notices . . . . .	v
<b>External SCM Tools (Windows)</b> . . . . .	1
External SCM Tools . . . . .	1
Preparing to Use an External SCM Tool (Windows) . . . . .	2
Setting Your SCM Connection Parameters (Windows) . . . . .	4
Adding Classes to the SCM Repository (Windows) . . . . .	4
Checking Classes Out from an SCM Repository (Windows) . . . . .	5
Checking Classes In with an External SCM Tool (Windows) . . . . .	7
Undoing Checkout with an External SCM Tool (Windows) . . . . .	8
Getting the Latest Version from an External SCM Tool (Windows) . . . . .	8
Comparing a Class to a Version in the SCM Repository (Windows) . . . . .	9
Displaying SCM History (Windows) . . . . .	9
Launching Your SCM Program (Windows) . . . . .	10



---

# Legal Notices

---

## Notices

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, USA.

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 IBM Canada Ltd., Department 071, 1150 Eglinton Avenue East, North York, Ontario M3C 1H7, Canada. Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

This publication may contain 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.

**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 the International Business Machines Corporation in the United States or other countries or both:

- AIX
- AS/400
- DB2

- CICS
- IBM
- OS/2
- OS/390
- RS/6000
- San Francisco
- VisualAge
- Visual Servlet
- WebSphere

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

Tivoli Management Environment, TME 10, and Tivoli Module Designer are trademarks of Tivoli Systems.

Encina and DCE Encina Lightweight Client are trademarks of Transarc Corporation.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the U.S. and/or other countries.

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

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks or registered trademarks of Intel Corporation in the U.S. and/or other countries.

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

---

## External SCM Tools (Windows)

The VisualAge for Java IDE offers an interface for checking .java source files in and out of an external software configuration management (SCM) system. This interface is a complementary feature that you can select when you install VisualAge for Java. It supports the following SCM tools:

- ClearCase 3.2 for Windows NT, from Rational Software Corporation
- PVCS Version Manager 6.0, from INTERSOLV, Inc.
- VisualAge TeamConnection Version 3.0, from IBM Corporation

---

## External SCM Tools

The interface from VisualAge for Java to external SCM tools uses Microsoft's Source Code Control (SCC) API. It is supported for Windows NT and Windows 95 clients. It may work with other SCC-compliant SCM tools, but IBM has only tested the products and releases listed above.

If you selected the interface to external SCM tools when you installed VisualAge for Java, the Workbench window provides a menu for adding classes to source control, checking classes in and out of the SCM repository, and importing the most recently checked-in version of a class from the SCM repository. Prior to VisualAge for Java, Version 2.0, if you wanted to check a class into your SCM tool's repository, you had to do the following steps:

1. Export the classes from the VisualAge for Java repository to the file system
2. Launch your SCM client program from outside the IDE
3. Use your SCM client program to check in the files that you created when you exported from VisualAge for Java

To check out a class, you would reverse the process. By contrast, with the interface to external SCM tools, the intermediate import and export steps are automated so you only need to select **Checkin** or **Checkout** from a menu in the IDE.

The interface from VisualAge for Java to external SCM tools does not provide any automatic synchronization of version names between the VisualAge for Java repository and the external SCM tool. VisualAge for Java does not prevent you from changing a program element in your workspace if you have omitted to check it out in the external SCM tool. The External SCM menu provides a convenient way for you to use an existing SCM tool without leaving the VisualAge for Java IDE, but you will need to correlate the functions of the two systems.

See the table below for a comparison of terms used by different SCM programs.

**[ENTERPRISE] External SCM Tools and 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 SCM 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 tool as your standard for application development.
- You have established practices for archiving applications on a particular enterprise server, for example for disaster recovery purposes.
- The repository in VisualAge for Java manages Java objects only; you may wish to manage all of your development artifacts with a single tool, 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 class 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.

### Comparison of SCM Terms

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

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

---

## Preparing to Use an External SCM Tool (Windows)

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

1. You must install the client code for your SCM tool on your workstation.
2. In the Windows NT registry, your SCM tool must be registered under the key **HKEY\_LOCAL\_MACHINE\SOFTWARE\SourceCodeControlProvider**, for the variable **ProviderRegKey**. To verify this, issue **regedit** from a command prompt. For more information, consult your SCM tool's documentation on enabling Source Code Control (SCC) functionality.

3. Using the native SCM software, someone has to create an organizational structure that your SCM tool will use to manage your .java files. Depending on the SCM software that you use, one or more of the following organizational units may be required:
  - VOBs and views (ClearCase)
  - Project files (PVCS Version Manager)
  - Families, releases, components, and work areas (VisualAge TeamConnection)

You will require this organizational information when you define your SCM connection.

4. 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 to 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 work directory that is recognized by your SCM tool. See below for a further explanation of work directories.
  - You are familiar with the organizational structures that your SCM tool uses.
  - You are familiar with settings that your SCM tool requires, such as drive mappings, file mounts, or the creation of views.
5. Some SCM tools, for example PVCS Version Manager, may need to be reconfigured to handle four-character (.java) file extensions.

Once you have met these prerequisites, you are ready to set the default parameters for VisualAge for Java to use when connecting to your external SCM tool. See below for a link to information on performing this task.

### The Work Directory

When you define the connection to your external SCM tool, you will need to specify a work directory. Here is an example of how the work directory is used.

If you were not using the automated interface from VisualAge for Java to your external SCM tool, 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 tool's repository.

With the SCM interface provided with VisualAge for Java, you can select HanoiApplet in the IDE, and then select **Tools > External SCM > Checkin** 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 work directory when you set your SCM connection parameters, the class is automatically exported to workdir\COM\ibm\ivj\examples\HanoiApplet.java.

As mentioned above in the list of prerequisites, the work directory must be recognized by your SCM tool.

---

## Setting Your SCM Connection Parameters (Windows)

Before you can use the VisualAge for Java interface for checking classes in and out of an external SCM tool, you must set your default SCM connection parameters. Before you can set your parameters, you must meet the prerequisites described in the file on preparing to use an external SCM tool. See below for a list of links to related information.

To establish the default parameters for your external SCM connection:

1. Select any VisualAge for Java project, package, or class in the Workbench window.
2. From the pop-up menu, select **Tools > External SCM > Change Connection**. The SCM Connection dialog box will open. (If you do not see this menu option, you may need to reinstall VisualAge for Java, this time selecting the complementary feature for External SCM Tools.)
3. Click **Change** to request a list of projects that your SCM tool recognizes.  
We use the term *project* to refer to your SCM tool's basic unit of organization. For more details, see the related files on SCM concepts and on preparing to use the external SCM interface. Links to these files are listed below.
4. Select a project from the list. (If you are using ClearCase, it does not actually matter which project you select from the list.) The SCM Connection dialog box will reappear, with the **Project Name** field completed.
5. Click **Browse** to select the default work directory that the SCM interface should use when it adds .java files to source control and checks them into the SCM repository. See the link to related concepts, below, for more information on work directories.
6. When you have returned to the SCM Connection dialog box and your selected directory appears in the **Work Directory** field, click **OK**.

These parameters will now be passed from VisualAge for Java to your external SCM tool, when you check classes in and out or perform other SCM operations.

The next step is to add your classes to source control in the external SCM tool, so you can check them in and out. See the list of links below for more information.

---

## Adding Classes to the SCM Repository (Windows)

Before you can use the check classes and interfaces in and out of an SCM tool, you must add them to source control in the SCM tool's repository. Before that, you must meet the prerequisites described in the file on preparing to use an external SCM tool. See the list of links below for more information.

In the VisualAge for Java IDE, you can select individual classes and interfaces that you wish to add to your SCM repository, as a preliminary step to checking them in and out. 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 repository at once. There is no other relationship between projects and packages in VisualAge for Java, and the organizational constructs in your SCM tool. You will need to correlate the names, contents, and versions of your VisualAge for Java program elements with the elements stored in your SCM repository.

To add classes and interfaces from VisualAge for Java to source control in the SCM repository:

1. From the Workbench window, select one or more projects, packages, classes, or interfaces.
2. From the pop-up menu, select **Tools > External SCM >Add**. VisualAge for Java will pass the list of classes and interfaces to the SCM tool, which validates the request.
3. If there are no problems, the Set Comments dialog box will appear. Enter the comments that you want the SCM tool to log when it adds the .java files to its repository, and click **OK**.
4. VisualAge for Java exports the classes and interfaces to the file system on your computer, in the work directory that you specified when you set your SCM connection parameters. Next, VisualAge for Java invokes the SCM client function to add those .java files to source control within the project that you specified when you set your SCM connection parameters.  
Wait for the Operation Complete dialog box to appear before you do any other software configuration tasks.
5. Click **OK** to close the Operation Complete dialog box.

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

As you create new classes and interfaces in your VisualAge for Java projects and packages, remember to add them to the SCM repository as well. There is no automated relationship between the VisualAge for Java source code repository and external SCM repositories.

To see where the classes and interfaces that you just added to the SCM repository are stored:

1. From the Workbench window, select one or more projects, packages, classes, or interfaces.
2. From the pop-up menu, select **Tools > External SCM >Launch SCM Program**.
3. Use the SCM tool's client software to check the contents of the SCM repository.

**[ENTERPRISE]** The team roles that determine privileges within VisualAge for Java do not apply to external SCM tools. Any member of the team can add any edition of a class or interface to the SCM repository, regardless of who owns that program element in VisualAge for Java. SCM activities are governed by the security policies of the SCM tool that you use.

---

## Checking Classes Out from an SCM Repository (Windows)

In the IDE, you can select classes and interfaces that you wish to check out of an external SCM tool's repository. When you take this action, VisualAge for Java automatically does the following things:

- Invokes your SCM client software to check out the most recently checked-in version of each class and interface
- Creates new open editions of the classes and interfaces in the VisualAge for Java repository
- Replaces the editions that were in your workspace with the new open editions

You can select classes and interfaces individually, or you can select them grouped in projects or packages as input to the SCM tool. There is no other relationship between projects and packages in VisualAge for Java and the organizational constructs used by your SCM tool.

Before you can check classes in or out, you must add them to source control in your SCM tools' repository. See the list of topics below for links to related information.

To check classes and interfaces out of your SCM tool's repository:

1. From the Workbench window, select the classes and interfaces that you wish to check out, or select the projects or packages that contain the classes you wish to check out.
2. From the pop-up menu, select **Tools > External SCM >Checkout**. VisualAge for Java will pass the list of classes and interfaces to the SCM tool, which validates the request.
3. If there are no problems, the Set Comments dialog box will appear. Enter the comments that you want the SCM tool to log when it checks the classes out of its repository, and click **OK**.

The classes and interfaces are checked out of the SCM repository, from the project that you identified when you set your SCM connection parameters. Open editions of the classes and interfaces are created in the VisualAge for Java repository, and added to your workspace. The classes and interfaces remain checked out of the SCM repository until you check them in again, or until you undo the checkout operation.

**[ENTERPRISE]** Since the class already existed in VisualAge for Java, the owner of the new edition is the existing class owner. As the person who created the open edition, you are the class developer and therefore the only person who can version that particular edition.

**[ENTERPRISE]** If you check out classes and interfaces from the SCM repository, and if the containing package or project in your VisualAge for Java workspace has been versioned, then a scratch edition of the package or project will be created. To prevent this, create an open edition of each project or package before selecting its classes for checkout.

### **Checking Out Classes that Were Not Developed in VisualAge for Java**

The above procedure requires that the classes or interfaces are already in your workspace, and therefore in the VisualAge for Java source code repository. To check out classes that exist in the SCM repository but not in the VisualAge for Java repository:

1. Start the native client software for your SCM tool by selecting **Tools >External SCM > Launch SCM Program** from the pop-up menu of any program element in the workspace.
2. Use the SCM client to check the classes out.
3. From the Workbench window, select **File > Import** to import the files into the VisualAge for Java repository. Open editions of the classes and interfaces will be created in the workspace.
4. **[ENTERPRISE]** To change ownership of the classes to the team member who will be responsible for them, select **Manage > Change Owner**.

---

## Checking Classes In with an External SCM Tool (Windows)

If you have checked classes or interfaces out of your SCM repository, you must either check them back in again or undo the checkout operation before anyone else can check them out.

In the VisualAge for Java IDE, you can select individual classes and interfaces that you wish to check into your SCM repository. 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 repository 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 workareas) in your SCM tool.

Before you can check classes in or out, you must add them to source control in your SCM tools' repository. See the list of topics below for links to related information.

To check classes and interfaces from VisualAge for Java into the SCM repository:

1. From the Workbench window, select one or more projects, packages, classes, or interfaces.
2. From the pop-up menu, select **Tools > External SCM > Checkin**. VisualAge for Java will pass the list of classes and interfaces to the SCM tool, which validates the request.
3. If there are no problems, the Set Comments dialog box appears. Enter the comments that you want the SCM tool to log when it checks the files into its repository, and click **OK**.
4. VisualAge for Java exports the classes and interfaces to the file system on your computer in the work file that you specified when you set your SCM connection parameters. Next, VisualAge for Java invokes the SCM client function to check those .java files into the SCM repository.

Wait for the Operation Complete dialog box to appear before you do any other software configuration tasks.

5. Click **OK** to close the Operation Complete dialog box.

Depending on the SCM tool that you use, you may see an error message that says the SCM operation failed when you try to check in files that have not changed. If you know that you have not changed a class, undo checkout instead of checking it in.

The exported classes and interfaces are checked into the SCM repository within the project that you specified when you set your SCM connection parameters. The classes and interfaces remain in your VisualAge for Java workspace and repository. If you do not check them out of your SCM repository again, VisualAge for Java will still allow you to create new open editions and to make changes, but your SCM tool will not allow you to check the changed editions in.

**[ENTERPRISE]** The team roles that determine privileges within VisualAge for Java do not apply to external SCM tools. Any member of the team can add any edition of a class or interface to the SCM repository, regardless of who owns that program element in VisualAge for Java. SCM activities are governed by the security policies of the SCM tool that you use.

---

## Undoing Checkout with an External SCM Tool (Windows)

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

To cancel 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 SCM >Undo Checkout**.
3. VisualAge for Java will pass the list of classes and interfaces to the SCM tool, which validates the request.
4. If there are no problems, the Operation Complete dialog box will appear. Click **OK**.

Other developers can now check the classes and interfaces out of the SCM repository. Depending on the external SCM tool that you use, a backup copy of the classes and interfaces may be created on your workstation.

---

## Getting the Latest Version from an External SCM Tool (Windows)

The interface to external SCM tools allows you to import the most recently checked-in editions of classes and interfaces from your SCM repository into VisualAge for Java, without checking them out. You might do this for testing purposes.

As with checkout, VisualAge for Java will do the following things when you get the latest classes and interfaces from your SCM library:

- Retrieves the most recently checked-in versions of the classes and interfaces, from the project that you specified when you set your SCM connection parameters, and copies them to the work directory that you specified.
- Automatically imports the class files from the work directory to the VisualAge for Java repository. New open editions are created.
- Replaces the editions that were in your workspace with the new open editions.

To get the most recently checked-in editions of classes and interfaces from your SCM repository:

1. From the Workbench window, select the classes and interfaces that you wish to retrieve, or select the projects or packages whose classes you wish to retrieve.
2. From the pop-up menu, select **Tools > External SCM >Get Latest**. VisualAge for Java will pass the list of classes and interfaces to the SCM tool, which validates the request.
3. If there are no problems, the Operation Complete dialog box will appear. Click **OK**.

Open editions of the classes and interfaces are created in the VisualAge for Java repository, and are added to your workspace.

**[ENTERPRISE]** Since the class already existed in VisualAge for Java, the owner of the new edition is the existing class owner. As the person who created the open edition, you are the class developer and therefore the only person who can version that particular edition.

**[ENTERPRISE]** If you get classes and interfaces from the SCM repository, and if the containing package or project in your VisualAge for Java workspace has been versioned, then a scratch edition of the package or project will be created. To prevent this, create an open edition of each project or package before you retrieve its classes.

### **Retrieving Classes That Do Not Exist in the VisualAge for Java Repository**

The above procedure requires that the classes or interfaces are already in your workspace, and therefore in the VisualAge for Java source code repository. To get the most recently checked-in versions of classes that exist in the SCM repository but not in the VisualAge for Java repository:

1. Start the native client software for your SCM tool by selecting **Tools > External SCM > Launch SCM Program** from the pop-up menu of any program element in the workspace.
2. Use the SCM client to get the .java files.
3. From the Workbench window, select **File > Import** to import the files into the VisualAge for Java repository. Open editions of the classes and interfaces will be created in the workspace.

---

## **Comparing a Class to a Version in the SCM Repository (Windows)**

After you have checked a class or interface out of an external SCM tool's repository, and made changes to it in VisualAge for Java, you may want to compare the edition of the class that is in your workspace to a previous one in the SCM tool's repository.

1. From the Workbench window, select the class or interface.
2. From the pop-up menu, select **Tools > External SCM > Compare**. VisualAge for Java passes the class name to the SCM tool for validation. If there are no problems, the class is automatically exported from the VisualAge for Java repository to the file system on your workstation, in the work directory that you specified when you set your SCM connection parameters. (This step is required by the SCM tools, which work with files rather than with the object-based repository of VisualAge for Java.)

If the automatic export step is successful, VisualAge for Java invokes the compare function of your SCM software program. If the export is not successful, confirm that you have checked the class out of the SCM repository. (If the file is checked in, the file status in the work directory is read-only.)

---

## **Displaying SCM History (Windows)**

In the VisualAge for Java IDE, you can request a history of SCM events for selected classes and interfaces. You can select projects or packages as a convenient way to see the SCM history for all of their contained classes and interfaces at once.

Adding classes to source control, checking in and out of the SCM repository, and undoing checkout are all examples of SCM events. To request a history:

1. From the Workbench window, select one or more projects, packages, classes, or interfaces.
2. From the pop-up menu, select **Tools > External SCM >History**. VisualAge for Java will pass the list of classes and interfaces to the SCM tool, which validates the request.
3. If there are no problems, the SCM client software on your workstation will be invoked to display a history for those classes and interfaces.

---

## Launching Your SCM Program (Windows)

VisualAge for Java provides a seamless interface from the IDE to several popular software configuration management tools. At times, you may also need to launch the native client software for your SCM tool, for example to check resource files into your SCM repository. (The SCM interface in VisualAge for Java only supports Java source files.)

To launch your SCM client software from inside the IDE:

1. From the Workbench window, select any project, package, classes, or interface.
2. From the pop-up menu, select **Tools > External SCM >Launch SCM Program**.

### Default SCM

If you have more than one SCM tool installed on your computer, VisualAge for Java automatically interfaces with the one that is defined as the default in the Windows registry, under the key

**HKEY\_LOCAL\_MACHINES\Software\SourceCodeControlProvider**, for the variable **ProviderRegKey**.