IBM VisualAge® for Java™, Version 3.5

Remote Access to Tool API

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. Remote Access to Tool API: Overview

Warning: Only experienced programmers should attempt to use Remote Access to Tool API. Remote Access to Tool API is ideally suited for use by ISVs.

Remote Access to Tool API and tool servlets provide a way for client applications to access the Tool Integrator APIs from outside the VisualAge for Java Integrated Development Environment (IDE) using the HTTP protocol. Remote Access to Tool API provides an HTTP daemon that supports tool servlets. A tool servlet is code you write that invokes the methods of the Tool Integrator API to perform a particular task.

To access the Tool Integrator APIs from outside the IDE, a client application sends an HTTP request to the tool servlet through the HTTP daemon. The daemon invokes the tool servlet. The tool servlet, in turn, invokes the methods of the Tool Integrator API. The API provides methods to perform the following tasks:

- Browse the VisualAge for Java IDE workspace and repository, and query objects
- Load editions from the repository into the workspace
- Import/export Java source, class files, and repository files
- Work with type (class/interface) source code
- Launch key IDE elements, such as various prompters and project/package/ type browsers
- Manipulate the CLASSPATH of your tool servlet
- Access tool data.

For more information about the Tool Integrator API, refer to the related concepts at the bottom of this page.

Tool servlets and the Java Servlet API
Tool servlets are a specialization of the Java Servlet API Version 2.0. They are specifically for execution within the VisualAge for Java IDE by tools that need to interact with the IDE from an outside process.
Developers familiar with the Java Servlet API should have no trouble writing tool servlets. From the developer's point of view, the only difference is in the names of the packages with which you need to work. The following table shows the package names for the Java Servlet API and tool servlets. Apart from these package names, and making use of the Tool Integrator API, writing a tool servlet is the same as writing a servlet.

Java Servlet API package name	Tool servlet package name
javax.servlet	com.ibm.ivj.toolserver.servletclasses.servlet
javax.servlet.http	com.ibm.ivj.toolserver.servletclasses.servlet.http
sun.servlet	com.ibm.ivj.toolserver.server.servlet
sun.servlet.http	com.ibm.ivj.toolserver.server.servlet.http

You can download documentation for the Java Servlet API from http://java.sun.com/products/servlet/. Be sure to request Version 2.0.

Accessing tool servlets
Remote Access to Tool API can be set so it starts automatically when the IDE

starts, or you can start it manually through an Options dialog in the IDE. Once Remote Access is started, the client application can access tool servlets in two ways:

- From a local host using the port number you specify in the Options dialog. The port number is stored in the following properties file, which the client application reads to obtain the port number:
  IBMVJavaRoot\ivjtools\tooldata\com-ibm-ivj-toolserver\toolport.properties
  where *IBMVJavaRoot* is the directory where VisualAge for Java is installed.
- From a remote host using one of the IP addresses or DNS names and port number you specify in the Options dialog. A remote host can determine the port number by reading the toolport.properties file.

For more information about setting Remote Access options, refer to the related tasks.

Why would you use Remote Access?
Remote Access provides a convenient method to perform basic tasks in VisualAge for Java, such as repository and workspace activities, from outside the IDE. For example, instead of using the IDE to back up the repository, you could write a tool servlet to perform this task, and invoke it from your application. Similarly, you could generate Java code using your application. Then, to test and debug the code in VisualAge for Java, you could invoke a tool servlet from your application that imports the source code into VisualAge for Java and starts an editor on the classes in your code.

Tool servlet samples
We provide samples that demonstrate how to code tool servlets that import Java files, export Java types, and launch a type browser in VisualAge for Java. Refer to the related tasks for details.

AIX

Required fileset
If Remote Access is running on AIX, the fileset IBMVJava.server.tool must be installed. You can specify this fileset when you install VisualAge for Java.

RELATED CONCEPTS

Overview of Tool Integrator API

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Setting Remote Access options
Starting and stopping Remote Access manually
Starting Remote Access automatically
Tracing the execution of Remote Access to Tool API
Accessing Remote Access to Tool API from a remote host
Placing tool servlets in the Remote Access directory
Passing parameters or assigning an alias to tool servlets
Using the tool servlet samples

Chapter 2. Using Remote Access to Tool API

Starting Remote Access automatically

The Options dialog is where you can specify that you want Remote Access to start automatically when the IDE starts.

To start Remote Access automatically, do the following:

1. In the VisualAge for Java Workbench, click **Windows > Options**. The Options dialog appears.
2. In the left-hand column of the Options dialog, click **Remote Access to Tool API**. The Remote Access to Tool API page of the Options dialog appears.
3. Select **Start Remote Access to Tool API on VisualAge startup**.
4. Click **OK**.

To prevent Remote Access from starting automatically, deselect the **Start Remote Access to Tool API on VisualAge startup** checkbox.

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlets

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Setting Remote Access options
Starting and stopping Remote Access manually
Tracing the execution of Remote Access to Tool API
Accessing Remote Access to Tool API from a remote host
Placing tool servlets in the Remote Access directory
Passing parameters or assigning an alias to tool servlets
Using the tool servlet samples

Starting and stopping Remote Access manually

The Options dialog is where you can start and stop Remote Access manually.

To start Remote Access, do the following:

1. In the VisualAge for Java Workbench, click **Windows > Options**. The Options dialog appears.
2. In the left-hand column of the Options dialog, click **Remote Access to Tool API**. The Remote Access to Tool API page of the Options dialog appears.
3. Click **Start Remote Access to Tool API**.
4. Click **OK**.

To stop Remote Access, click **Stop Remote Access to Tool API**.

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlets

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Setting Remote Access options
Starting Remote Access automatically
Tracing the execution of Remote Access to Tool API
Accessing Remote Access to Tool API from a remote host
Placing tool servlets in the Remote Access directory
Passing parameters or assigning an alias to tool servlets
Using the tool servlet samples

Setting Remote Access options

The Options dialog is where you specify options for Remote Access, and where you can start and stop Remote Access manually.

To set Remote Access options, do the following:

1. In the VisualAge for Java Workbench, click **Windows > Options**. The Options dialog appears.
2. In the left-hand column of the Options dialog, click **Remote Access to Tool API**. The Remote Access to Tool API page of the Options dialog appears.
3. Make your selections in the Options dialog as follows:
   - **Start Remote Access to Tool API on VisualAge startup:**Start Remote Access automatically when the IDE starts.
   - **Enable tracing of Remote Access to Tool API:**Trace the execution of Remote Access to Tool API and the executing tool servlet. This includes any calls made to standard out and standard err in the executing tool servlet. For example, a statement such as System.out.println("This is a trace message") in the tool servlet will be displayed when tracing is turned on. Trace output is displayed in the Console window.
   - **Use system-generated port:**Start Remote Access using a port number chosen by the system. This number might be different each time Remote Access is started.
   - **Use user-defined port:**Start Remote Access using a specific port number instead of letting the system generate one for you. This is useful if you will allow remote hosts to access tool servlets, and want to define a known port number. If the user-defined port is occupied by another process, Remote Access will not start. The valid range for the port number is 1025-65534.
   - **Start Remote Access to Tool API:**Start Remote Access manually using the options currently specified on the Options page. If Remote Access is already started, this button is grayed out.
   - **Stop Remote Access to Tool API:**Stop Remote Access manually. If Remote Access is already stopped, this button is grayed out.
   - **Access permitted from the following hosts:** Specify the IP address or DNS name of the remote host that will access the Remote Access server. To specify several hosts, separate the IP addresses or DNS names with a space. The default is to access the Remote Access server using the local host (127.0.0.1).
4. Click **OK**.

If you want to restore default settings, click **Defaults**. If you want to apply changes without exiting the window, click **Apply**.

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlets

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Starting and stopping Remote Access manually
Starting Remote Access automatically
Tracing the execution of Remote Access to Tool API
Accessing Remote Access to Tool API from a remote host
Placing tool servlets in the Remote Access directory
Passing parameters or assigning an alias to tool servlets
Using the tool servlet samples

Accessing Remote Access to Tool API from a remote host

The default is to access Remote Access from the local host. You can also access Remote Access from a remote host by specifying the IP address or DNS name of the remote host in the Options dialog.

To set up remote access, do the following:

1. In the VisualAge for Java Workbench, click **Windows > Options**. The Options dialog appears.
2. In the left-hand column of the Options dialog, click **Remote Access to Tool API**. The Remote Access to Tool API page of the Options dialog appears.
3. In the **Access permitted from the following hosts**field, enter the IP address or DNS name of the remote host. To specify several hosts, separate the IP addresses or DNS names with a space.
4. Click **OK**.

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlets

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Setting Remote Access options
Starting and stopping Remote Access manually
Starting Remote Access automatically
Tracing the execution of Remote Access to Tool API
Placing tool servlets in the Remote Access directory
Passing parameters or assigning an alias to tool servlets
Using the tool servlet samples

Compiling and testing tool servlets using VisualAge for Java

This section shows you how to set up VisualAge for Java so you can use it to compile and test your tool servlets.

Setting up for compiling
To set up VisualAge for Java for compiling, do the following:

1. Add the IBM IDE Utility Class Libraries feature to VisualAge for Java by doing the following:
   a. In the Workbench, click **File > Quick Start**. The Quick Start window opens.
   b. In the left-hand pane of the Quick Start window, click **Features**, and in the Features pane, click **Add Feature**. Click **OK**. The Selection Required window opens.

   c. Click on **IBM IDE Utility Class Libraries**, and then click **OK.**
2. To compile and debug your tool servlets, you'll need certain classes contained in packages we provide.
   Create a new project to contain the packages that contain those classes. In the Projects page of the Workbench, click **Selected > Add > Project**. Specify the name of the new project, such as ToolServer, in the Add Projects window.
3. Import into the project you created the required packages by doing the following:
   a. Select the project in the Projects page.
   b. Click **File > Import**. The Import window opens.
   c. Click **Directory**, and then click **Next**.
   d. In the Import from a Directory page, specify the following directory:
      IBMVJavaRoot\ide\TOOLS\com-ibm-ivj-toolserver\com
      where *IBMVJavaRoot* is the directory where VisualAge for Java is installed.
   e. Click **.class**. Ensure **.java**or **resource** are not selected.
   f. Click **Finish**.

VisualAge for Java is now set up to compile your tool servlets.

You should create another project to contain your tool servlets. Once you create the project, you can code new servlets, or you can use the samples we provide.

Importing the tool servlet samples
To use the samples we provide, import them into the project you created for your tool servlets by doing the following:

1. In the Projects page, select the project for your tool servlets.
2. Click **File > Import**. The Import window opens.
3. Click **Directory**, and then click **Next**.
4. In the Import from a Directory page, specify the following directory:
   IBMVJavaRoot\ide\TOOLS\com-ibm-ivj-
   toolserver\servlets\code\com\ibm\ivj\
   toolserver\samples
   where *IBMVJavaRoot* is the directory where VisualAge for Java is installed.
5. Click **.class** and **.java.**
6. Click **Finish**.

Setting up for testing
To set up VisualAge for Java for testing, do the following:

1. If Remote Access to Tool API is running, stop it. To test your servlets, Remote Access needs to be running within the IDE. To stop Remote Access, refer to "Stopping and starting Remote Access manually" in the related tasks at the bottom of this page.
2. In the Packages page of the Workbench, expand the com.ibm.ivj.toolserver package.
3. Click on the ToolHttpServer class, with mouse button 2. From the pop-up menu, click **Properties**.
4. In the Properties window, click the **Program** tab.
5. In the **Command line arguments** field of the Program page, enter the following arguments:
   -v -p *portnumber* -d .
   where *portnumber* is the number of the port to start Remote Access. The valid range for *portnumber* is 1025-65534.

6. Click the **Class Path**tab.
7. In the Class Path page, click the **Edit** button for **Project path**. The Class Path window opens.
8. In the Class Path window, select the project the servlets are in, and the **IBM IDE Utility Class Libraries** feature. Click **OK**.
9. In the Class Path page, click the **Edit** button for **Extra directories path**. The VisualAge window opens. Click **Add directory**. The Select a directory window opens.
10. Select the following directory: ide\project_resources\IBM IDE Utility local implementation. Click **OK**.
11. Repeat the previous two steps to specify the following two directories: IBM IDE UI class libraries and IBM IDE Utility local implementation.
12. Use the Class Path window to specify any projects that contain tool servlets you want to run. Remote Access can only execute servlets that are on its CLASSPATH.
13. Click **OK** until all the class path windows are closed.
14. Click on the ToolHttpServer class, and then click the run icon. Remote Access to Tool API starts running within the IDE.

Testing your tool servlets
You can invoke your tool servlets from a Web browser by entering a URL like this:
http://localhost:*portnumber*/servlet/*package.toolservlet*
where *portnumber* is the number of the port you specified in Setting up for testing (page 6), and *package.toolservlet* is the name of the tool servlet and the package it is in.

In VisualAge for Java, you can use the debugger to test your servlets by setting breakpoints, and by stepping through the code to make changes.

Once you finish testing your tool servlets, you should stop Remote Access from running within the IDE. To stop it, select ToolHttpServer in the Console or debugger window, and click the terminate icon.

Deploying your tool servlets
Once your tool servlets are compiled and running without errors, you must deploy the compiled (.class) versions of your tool servlets to the servlets directory in the following path:
IBMVJavaRoot\ide\TOOLS\com-ibm-ivj-toolserver\servlets

where *IBMVJavaRoot* is the directory where VisualAge for Java is installed. When your servlets are in the servlets directory, they are available to Remote Access to Tool API.

To deploy your tool servlets, do the following:

1. In the Projects page, select the project for your tool servlets.
2. Click **File > Export**. The Export window opens.
3. Click **Directory**, and then click **Next**.
4. In the Export to a Directory page, specify the following directory:
   IBMVJavaRoot\ide\TOOLS\com-ibm-ivj-toolserver\servlets
   where *IBMVJavaRoot* is the directory where VisualAge for Java is installed.
5. Click **.class**.
6. If you have any resource files, click **resource** and the **Details** button to specify the resources you want to deploy.

7. Click **Finish**.

Once your tool servlets are deployed, you can invoke them and pass parameters to them from your browser. Refer to “Passing parameters and assigning an alias to tool servlets” in the related tasks for details. To invoke your tool servlets, Remote Access must be running. For details about starting Remote Access, refer to the related tasks.

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlets

RELATED TASKS

Setting Remote Access options
Starting and stopping Remote Access manually
Starting Remote Access automatically
Tracing the execution of Remote Access to Tool API
Accessing Remote Access to Tool API from a remote host
Passing parameters or assigning an alias to tool servlets
Using the tool servlet samples

Placing tool servlets in the Remote Access directory

You must place the compiled (.class) versions of your tool servlets in the following path:

IBMVJavaRoot\ide\TOOLS\com-ibm-ivj-toolserver\servlets

where *IBMVJavaRoot* is the directory where VisualAge for Java is installed.

Once your servlets are in the servlets directory, they are available to Remote Access to Tool API.

You can also pass parameters and assign an alias to tool servlets. Refer to the related tasks for details.

To deploy your tool servlets in the Remote Access directory using VisualAge for Java, refer to “Compiling and testing tool servlets using VisualAge for Java” in the related tasks.

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlets

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Setting Remote Access options
Starting and stopping Remote Access manually
Starting Remote Access automatically
Tracing the execution of Remote Access to Tool API
Accessing Remote Access to Tool API from a remote host
Passing parameters or assigning an alias to tool servlets
Using the tool servlet samples

Passing parameters or assigning an alias to tool servlets

You can pass parameters or assign an alias to tool servlets.

Passing Parameters
You can pass parameters to tool servlets in two ways:

- By including the parameter when you invoke the tool servlet.

  To pass a single parameter, use the ? (question mark) character like this:

  ```
  http://localhost:34567/servlet/TestServlet?path=d:\importdir
  ```

  To pass multiple parameters, separate them with the & (ampersand) character like this:

  ```
  http://localhost:34567/servlet/TestServlet?path=d:\importdir&project=Test_Project
  ```

  Note: If the parameter you pass contains a relative directory name, the starting directory is *IBMVJavaRoot*\ide\program. For example, if you specify path=d:..\importdir, the full path is path=d:*IBMVJavaRoot*\ide\program\importdir, where *IBMVJavaRoot* is the directory where VisualAge for Java is installed.
- When Remote Access starts. Enter the following statements to the servlet.properties file:

  ```
  # servlet.properties
  # Define servlets here
  # <servletname>.code=<servletclass>
  # servlet.<servletname>.initArgs=<comma-delimited list of {name,value}
  pairs that can be accessed by the servlet using the servlet API calls>
  ```

  For example, if you want to pass type and project parameters to LaunchBrowserServlet, you would enter the following statements:

  ```
  # servlet.properties
  # Define servlets here
  # servlet.test.code=com.ibm.ivj.toolserver.samples.LaunchBrowserServlet
  # servlet.test.initArgs={type,java.net.Socket},{project,Java class
  libraries}
  ```

  The servlet.properties file is in the following path:

  ```
  IBMVJavaRoot\ide\TOOLS\com-ibm-ivj-toolserver\servlets
  ```

  where *IBMVJavaRoot* is the directory where VisualAge for Java is installed.

Assigning an Alias
You might want to assign an alias to one of your servlets to simplify the invocation of the servlet. For example, if you have a servlet called HelloWorldToolServlet, but you want to invoke it by referring to it as hi, you can add the following statement to servlet.properties :

```
servlet.hi.code=com.ibm.ivj.toolserver.samples.HelloWorldToolServlet
```

You can then invoke the servlet using its alias like this:

```
http://localhost:34567/servlet/hi
```

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlet

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Setting Remote Access options

Starting and stopping Remote Access manually
Starting Remote Access automatically
Tracing the execution of Remote Access to Tool API
Accessing Remote Access to Tool API from a remote host
Placing tool servlets in the Remote Access directory
Using the tool servlet samples

Tracing the execution of Remote Access to Tool API

You can trace the execution of Remote Access and the executing tool servlet by viewing trace output in the Console window of the IDE. This includes any calls made to standard out and standard err in the executing tool servlet.

To start the trace, do the following:

1. In the VisualAge for Java Workbench, click **Windows > Options**. The Options dialog appears.
2. In the left-hand column of the Options dialog, click **Remote Access to Tool API**. The Remote Access to Tool API page of the Options dialog appears.
3. Select **Enable tracing of Remote Access to Tool API**.
4. Click **OK**.

The Console window will be opened automatically when Remote Access is started, and will be cleared automatically when Remote Access is stopped.

To stop tracing, deselect **Enable tracing of Remote Access to Tool API**.

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlets

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Setting Remote Access options
Starting and stopping Remote Access manually
Starting Remote Access automatically
Accessing Remote Access to Tool API from a remote host
Placing tool servlets in the Remote Access directory
Passing parameters or assigning an alias to tool servlets
Using the tool servlet samples

Using the tool servlet samples

We provide the following samples that demonstrate how to code tool servlets that access the Tool Integrator APIs from outside the IDE:

- **ExportFileServlet.java:** exports a type from VisualAge for Java to the file system.
- **ImportFileServlet.java:**imports Java files into VisualAge for Java from the file system.
- **LaunchBrowserServlet.java:** launches a type browser in VisualAge for Java.

The samples are in the following path:

IBMVJavaRoot\ide\TOOLS\com-ibm-ivj-toolserver\servlets\code\com\ibm\ivj\toolserver\samples

where *IBMVJavaRoot* is the directory where VisualAge for Java is installed.

The file *IBMVJavaRoot*\ide\TOOLS\com-ibm-ivj-toolserver\CallTestServlets.html contains more details about the samples, including usage information, and provides a user interface you can use to test the samples.

To use the samples with VisualAge for Java, refer to “Compiling and testing tool servlets using VisualAge for Java” in the related tasks.

RELATED CONCEPTS

Overview of Remote Access to Tool API and tool servlets

RELATED TASKS

Compiling and testing tool servlets using VisualAge for Java
Setting Remote Access options
Starting and stopping Remote Access manually
Starting Remote Access automatically
Tracing the execution of Remote Access to Tool API
Accessing Remote Access to Tool API from a remote host
Placing tool servlets in the Remote Access directory
Passing parameters or assigning an alias to tool servlets

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:
© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 1997, 2000. All rights reserved.

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.