



IBM WebSphere Business Components Studio

Installation Guide

Version 1.1

... A member of the WebSphere Business Components family

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 27.

First Edition (November 2000)

This edition applies to version 1.1 of IBM WebSphere Business Components Studio (product number 5639-M22), and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Corrections and suggestions for future revisions of this document are appreciated. Mail your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/KB7/1150/TOR
1150 Eglinton Avenue East
Toronto, Ontario, M3C 1H7
Canada

When you send information to IBM, you grant to IBM a nonexclusive right to use or distribute the information in any way they believe appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2000. All rights reserved.

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

Contents

IBM WebSphere Business Components Studio, Version 1.1 installation.....	1
General installation instructions.....	1
Hardware requirements.....	1
Software requirements.....	2
Installing the product from the CD.....	4
Set up a development environment	5
Prepare the VisualAge for Java environment.....	5
ACServices Rose model.....	7
Install an AC .cat file in Rational Rose.....	7
Tools.....	9
Advanced Component Services.....	11
AC Services prerequisites and files.....	11
Deploy the AC Services in the VisualAge for Java WebSphere Test Environment	12
Configure VisualAge for Java for AC Services.....	12
Import third-party and AC Services files into projects	12
Import AC Services EJB .jar files into EJB groups	13
Configure AC Services	14
Run the System Management Console.....	14
Deploy the AC Services in WebSphere Application Server	14
Add .jar files to the server's dependent classpath.....	14
Add .jar files to the command line argument of the server's classpath	14
Deploy AC Services EJBs in WebSphere Application Server.....	15
Configure the JNDI database	16
Add .jar files to the client's classpath.....	16
Add .jar files for the AC System Management Console's classpath	16
Configure the AC Services.....	17
Configure the System Management Console.....	17
Advanced Components	18
Install and deploy Advanced Components in WebSphere	18
Create the database and its tables.....	18
Configure the datasources in the application server.....	19
Configure the AC by making changes to its configuration file.....	19
Run the AC Deployment Tool	20
Deploy and register the AC in WebSphere	20
Install, deploy, and run an AC in VisualAge for Java.....	20
Import the ACs and their dependencies	21
Import and deploy the EJBs of the AC	21
Run the AC	22
Samples and examples	23
Help system	24

Install the Help System	24
Run the Help System.....	24
Configure the Studio help system to run on a Windows server	24
Help System Troubleshooting	25
Help System does not launch	25
Help System for VisualAge for Java or DB2 does not run	25
Other problems.....	25
Notices.....	27
Trademarks and service marks	28

IBM WebSphere Business Components Studio, Version 1.1 installation

The green text represents changes that were made to this document after the WebSphere Business Components Studio Version 1.1 CD was created.

General installation instructions

IBM WebSphere Business Components Studio, Version 1.1 provides a set of standard EJB-based components and associated development tools that help you build, manage, and deploy e-business applications.

To successfully install this product on your Microsoft® Windows NT system, make sure you have the proper hardware and software. Basic software requirements vary with your usage. This table provides some broad installation requirements for your specific usage:

<i>General installation requirements</i>	
Usage description	Software requirements
To install the complete product	Windows NT (4.0 and Fixpack 6a) and WebSphere® Application Server Advanced Edition (3.5 and Fixpack 2)
To run the samples	DB2 (DB2 UDB V6.1 and Fixpack 4) and WebSphere Application Server Advanced Edition (3.5 and Fixpack 2)
To develop applications	Refer to Software requirements, below.

Hardware requirements

The minimum hardware requirements include the following:

- 450 Mhz Pentium III system
- 512 Mb of RAM
- 6 Gb of available hard drive space
- CD ROM

Software requirements

The following table lists all products (both IBM and non-IBM) that are needed at installation, development, and deployment:

<i>Installation, development, and deployment software requirements</i>			
Product Name	Description	Version/Release	Components that need it
Compiler and development tools			
VisualAge® for Java™ (VAJ) Enterprise Edition	<p>Integrated Development Environment for building and testing Java applets, servlets, and Enterprise JavaBean components.</p> <p>Provider: IBM</p> <p>How to obtain: http://www-4.ibm.com/software/ad/vajava/</p>	3.5	All
Rational Rose®	<p>Use this application to translate high level business requirements into an architecture that generates component code in your chosen language.</p> <p>Provider: Rational</p> <p>How to obtain: http://www.rational.com/products/rose/index.jsp</p>	2000e Modeler Edition or Enterprise Edition	AC + BC
MQ Series based product	<p>The IBM MQSeries family provides an open, scalable, industrial-strength messaging and information infrastructure, enabling enterprises and beyond to integrate business processes.</p> <p>Provider: IBM</p> <p>How to obtain: http://www.ibm.com/software/ts/mqseries/v5/</p>	5.1	AC Event Services
MA0C: MQSeries - Publish/Subscribe	<p>MQSeries Publish/Subscribe helps you distribute information to where it is wanted.</p> <p>Provider: IBM</p> <p>How to obtain: http://www.ibm.com/software/ts/mqseries/txppacs/ma0c.html</p>	5.1	AC Event Services

Installation, development, and deployment software requirements

Product Name	Description	Version/Release	Components that need it
MA88: MQSeries classes for Java and MQSeries classes for Java Message Service	<p>The MQSeries classes for Java allow a program written in the Java programming language to connect to MQSeries as an MQSeries client using TCP/IP, or directly to an MQSeries server using the Java Native Interface (JNI). They allow Java applets, applications, and servlets access to the messaging and queuing services of MQSeries. If the client-style connection is used, no additional MQSeries code is required on the client machine. The MQSeries classes for Java enable a message-based approach to application integration using Java.</p> <p>MQSeries classes for Java Message Service is a set of Java classes that implement Sun Microsystem's Java Message Service specification. A JMS application can use the classes to send MQSeries messages to either existing MQSeries or new JMS applications.</p> <p>Provider: IBM</p> <p>How to obtain: http://www.ibm.com/software/ts/mqseries/txppacs/ma88.html</p>	5.1	AC Event Services
LDAP	<p>A JNDI (Java Naming and Directory Interface) service.</p> <p>Provider: IBM</p> <p>How to obtain: http://www.ibm.com/software/network/directory/downloads/</p>	3.2	AC Event Services
Database			
DB2 [®]	<p>A relational database management system. Use this for development and deployment.</p> <p>Provider: IBM</p> <p>How to obtain: http://www-4.ibm.com/software/webervers/download.html</p>	DB2 UDB V6.1 and Fixpack 4	
Oracle [®] 8i	<p>A relational database management system. Use this for deployment.</p> <p>Provider: Oracle</p> <p>How to obtain: http://www.oracle.com/database/oracle8i/</p>	8.1.6.1	

Installation, development, and deployment software requirements

Product Name	Description	Version/Release	Components that need it
OS / App server / EJB server			
Windows NT Server	Provider: Microsoft How to obtain: http://www.microsoft.com/ntserver/	4.0, service Fixpack 6a or greater	
WebSphere Application Server Advanced Edition Comes with JSDK, IBM HTTP Server, XML4J, JNDI	Provider: IBM How to obtain: http://w3.software.ibm.com/webservers/html/downloads.html	3.5 Fixpack 2	
Web browser (for development time and samples)	Netscape 4.7 or higher, or Internet Explorer 5.0 or higher Netscape provider: Netscape How to obtain: http://www.netscape.com/download Internet Explorer provider: Microsoft How to obtain: http://www.microsoft.com/downloads		Documentation (Online Help System)

Installing the product from the CD

To begin installation, do the following:

1. Insert the CD in your CD drive.
2. Use Windows Explorer to navigate to the CD's directory. Move to **WSBCInstall** directory and run the **Setup.exe** file. This initializes the installation process. Read through each window at your own rate, following the instructions.
3. When you get to the **Select Installation Type** dialog, you have a choice between a **Typical** or **Custom** install. It is recommended that you choose the **Typical** install, which installs the components, samples, tools, and documentation. You have a choice of where (drive/directory) you want them to be installed.

Set up a development environment

This section describes how to prepare the VisualAge for Java environment, and how to install into Rational Rose the .cat files included in the Studio CD. This section also has a description of the Rose model for the AC services.

Prepare the VisualAge for Java environment

The following instructions describe how to prepare VisualAge for Java so that you can deploy and run an AC in VisualAge for Java.

1. Add required features to VisualAge for Java:
 - a. From the Workbench, select **File > Quick Start**.
 - b. In the **Quick Start** dialog, select **Features** on the left and **Add Feature** on the right. Click **OK**.
 - c. From the list of available features, select the following:
 - **IBM EJB Development Environment**
 - **IBM WebSphere Test Environment**
 - d. Click **OK**.
2. Set the Workspace classpath:
 - a. On the menu bar, click **Window > Options**
 - b. Click **Resources** on the tree
 - c. Edit the Workspace classpath to include the following paths:
 - `<VisualAge for Java Root>\ide\project_resources\IBM XML Parser for Java\`
 - `<VisualAge for Java Root>\ide\project_resources\IBM Enterprise Extension Libraries\`
 - `<VisualAge for Java Root>\ide\project_resources\IBM WebSphere Test Environment\`
3. Add db2java.zip to the Workspace classpath:
 - a. From the Workbench, select the **Window** menu and then select **Options**.
 - b. On the left side of the dialog, click **Resources**.
 - c. On the right side of the dialog, enter the full path (including the drive) to the db2java.zip file that came with your DB2 installation. To get the path to the file by browsing for it, click **Edit**. In the dialog that appears, click **Add Jar/Zip** and navigate to the db2java.zip file. Click **Open** when you have selected the file and then click **OK**.
 - d. Click **Apply** and then click **OK**.

4. Configure VisualAge for Java to use DB2 instead of InstantDB:
 - a. Create a new database in DB2. Assume that you have created a database named NAME_SERVER.
 - b. From the Workbench, select the **WorkSpace** menu and then select **Tools** and then **Launch WebSphere Test Environment**.
 - c. In the WebSphere Test Environment Control Center, select the **Persistent Name Server** node.
 - d. In the right-hand pane, fill in the following fields.

Field	Value
Database driver	COM.ibm.db2.jdbc.app.DB2Driver
Database URL	jdbc:db2:NAME_SERVER

- e. Click **Apply**. Click **Start Name Server**.
- f. In DB2, verify that the NAME_SERVER database now has the following tables:
 - BINDINGBEANTBL
 - CONTEXTBEANTBL
 - PROPERTYBEANTBL

A common problem you may encounter is that the AC Command Service uses JNDI to register the command properties, using as the name *<AC instance name + function name + properties>*. This often exceeds the default length for JNDI names, which is 64 characters. To use JNDI names that are longer than 64 characters, you have to alter the naming server's tables in the database (or the VisualAge for Java database if it is different from the one use in the WebSphere deployment) to a length that can handle the JNDI name. In most circumstances, a length of 128 is sufficient. To do this for DB2, open a DB2 command line processor and type the following commands, replacing *databaseName* with the name of the database you are using (WAS for WebSphere). This example makes max length 128.

```
CONNECT TO databaseName
ALTER TABLE EJSADMIN.BINDINGBEANTBL ALTER COLUMN NAME
SET DATA TYPE VARCHAR (128)
ALTER TABLE EJSADMIN.PROPERTYBEANTBL ALTER COLUMN
NAME SET DATA TYPE VARCHAR (128)
ALTER TABLE EJSADMIN.PROPERTYBEANTBL ALTER COLUMN
PROPERTYNAME SET DATA TYPE VARCHAR (128)
DISCONNECT databaseName
```

- g. Select the **DataSource Configuration** node

- h. For each datasource that you wish to configure VisualAge for Java to use, provide values for the following fields (the example values are from the Customer Profile AC):

Field	Value
DB name	cpBoth
DB driver	COM.ibm.db2.jdbc.app.DB2Driver
DB URL	jdbc:db2:cpBoth
DB type	JDBC

Note: If you are configuring multiple Customer Profile datasources, set the DB type to JTA for each of the Customer Profile datasources. Otherwise, set the DB type of JDBC.

See "Create the database tables" in the Advanced Component section for the names of the databases used by the Advanced Components included on the Studio CD.

ACServices Rose model

The Rose model of ACServices (Services/Base Services/rose/ACServices/acs.mdl) provides general design information about each service included in ACServices. This information is contained in the Logical View folder of the model.

The Logical View folder of ACServices model contains the following elements:

- The acs category, which contains a subcategory for each of the services that comprise the ACServices layer. These services are:
 - AC Command
 - AC Context
 - AC Event
 - AC Exception
 - AC NLS
 - AC System Management
- The Main diagram, which shows dependency relationships between the services. For simplicity, dependencies that are always presumed to exist, such as a dependency on Java, are not shown.

The subcategory for each service contains the classes and interfaces that comprise the service. In addition, each service has a diagram that shows the detail about each class and interface and the relationships between them.

Install an AC .cat file in Rational Rose

To view the model of the interface of an AC, you must load the appropriate .cat file. Here are the instructions for loading the .cat files for a specific component:

1. Open Rational Rose
2. Go to **File > Edit Path Map ...**

3. In the Virtual Path Map dialog, type the following into the **Symbol:** text box

WSBC_ROOT

In the **Actual Path:** text box, type the drive and path to the ACFeatures\Components directory of the installation directory.

4. Click **Add** and then click **Close**.
5. From the **File** menu, select **Units** and then select **Load...**
6. From the Load Category From dialog, choose the appropriate .cat file:

AC or parameter suite	.cat file
Customer Profile	CustomerProfileAC.cat
Product Catalog	ProductCatalogAC.cat
Common Customer Parameter Suite	CommonCustomerInterfaceSuite.cat
Common Parameter Suite	CrossDomainInterfaceSuite.cat

7. The files are located in the rose directory of the component you want to load up (for example, *<Drive>*:\Program Files\IBM\WSBC\1.1\ACFeatures\Components\ProductCatalog\rose\ProductCatalogAC.cat)
8. This installs the main .cat file for the component. Any subunit .cat files can be loaded by right-clicking the unit and then selecting **Unit** and then selecting **Load...** Rose automatically finds the correct subunit files for Customer Profile and Product Catalog. You will have to browse for the subunits of the Common Parameter Suite and the Common Customer Parameter Suite.

Tools

Provided the system has the prerequisite version of WebSphere Application Server, VisualAge for Java, and Rational Rose, the installation automatically makes all the necessary configurations for the WSBC Code Generator and most of the configuration required for the AC Deployment Tool. This section describes what you have to do to complete the configuration of the AC Deployment Tool.

You can run the AC Deployment tool in the following three ways, each of which requires a specific set of configuration actions to be completed:

- As a standalone application by entering a command prompt
- From the VisualAge for Java menu
- From VisualAge for Java IDE

To configure AC Deployment Tool to run from a command line, complete the following steps:

1. Edit the `actools.ini` file found in `<WSBCROOT>/tools` so that the `basepath` is set to the directory where `settings.xml` is found. The `basepath` should have double slashes as delimiters, and should end in double slashes, as in the following example:

```
basepath=c:\\Program Files\\ibm\\WSBC\\1.1\\tools\\
```
2. Open a command prompt.
3. Run the `setupCmdLine.bat` file found in the `<WebSphereRoot>\AppServer\bin` directory.
4. The environment variable `WSBCTOOLS_PATH` should be set by the installation. If not, set this environment variable to the root of the `tools` directory in `WSBC`.
5. Run `setWSenv.bat` from a command line (You only need to do this once per command session).
6. To run the AC Deployment Tool, type `runDeploymentTool.bat` on a command line.

The tool will work when either WebSphere or the WebSphere Test environment is running.

To configure AC Deployment Tool to run from the VisualAge for Java Tool Menu, complete the following steps:

1. Start VisualAge for Java.
2. In VisualAge, you must set the `WorkSpace` classpath:
 - a. On the menu bar, click **Window** and then click **Options**.
 - b. Click **Resources** on the tree.
 - c. Edit the `Workspace` classpath to include the following paths:
 - `<VAJROOT>\ide\project_resources\IBM XML Parser for Java\;`
 - `<VAJROOT>\ide\project_resources\IBM Enterprise Extension Libraries\;`
 - `<VAJROOT>\ide\project_resources\IBM WebSphere Test Environment\;`
3. Click **Apply**.
4. Click **OK**.

You should now be able to run the tool from the VisualAge for Java menu bar. To do so, click **Workspace**. Click **Tools** and then click **AC Deployment Tool**.

To configure the AC Deployment Tool to run from the VisualAge for Java IDE, complete the following steps:

1. In the VisualAge for Java Workbench, import ACModel.jar and ACToolsDeployment.jar from the <WSBCROOT>\tools\lib directory into the project(s) you want to use.
2. Import actools.ini and settings.xml found in <WSBCROOT>\tools as a resource for the project. You can do this by copying the files in Windows Explorer, and pasting them to the resource folder of your project, located in the <VAJRoot>\ide\project_resources directory.
3. Edit the actools.ini file in the resources directory so that basepath is set to the resource directory where settings.xml is found. The basepath should have double slashes as delimiters, and should end in double slashes, as in the following example:

```
basepath=c:\\Program Files\\ibm\\WSBC\\1.1\\tools\\
```
4. Right-click on com.ibm.wsbc.actools.deployment.DeployerTool.
5. In the context menu, select **Run**, and then click **Check Classpath**.
6. Edit the project path to include the following:
 - o All projects where the .jar files listed at the top were imported to
 - o IBM Enterprise Extension Libraries
 - o IBM WebSphere Test Environment
 - o IBM XML Parser for Java
7. **Click OK**.
8. Right-click on com.ibm.wsbc.actools.deployment.DeployerTool.
9. In the context menu, select **Run**, and then click **Run main**.

Once you start the AC Deployment Tool, you must specify the root directory for WebSphere, in order to deploy ACs to WebSphere. To specify the root directory for WebSphere, complete the following steps:

1. Run the AC Deployment Tool.
2. On the **Actions** menu, click **Set Preferences**.
3. In the **Options** dialog that appears, select **WebSphere** as your EJB Server.
4. In the **Root Directory** field, type the path to the directory where WebSphere is installed on your system.

Advanced Component Services

The following instructions show you how to install and deploy the AC services on the VisualAge for Java WebSphere Test Environment and WebSphere Application Server.

AC Services prerequisites and files

These instructions assume that you have installed IBM WebSphere Business Components Studio in the directory *<WSBC directory>*, WebSphere Application Server 3.5 Advanced Edition (WAS) in *C:\WebSphere\AppServer*, and VisualAge for Java in *C:\VAJava35* with the following features:

- IBM EJB Development Environment 3.5
- IBM WebSphere Test Environment 3.5

You can find the following files in the directory *<WSBC directory>\<AC Services subdirectory>*, where *<AC Services subdirectory>* is *Services/BaseServices*:

File	Description
ACServicesEJB.jar ACServicesEJBClient.jar ACServicesEJBDeployed.jar	Contains command and system management EJBs
ACServicesClient.jar ACServicesServer.jar	Contains the non-EJB classes for the client side and server side of the AC services
ACEI18NEJB.jar ACEI18NEJBClient.jar ACEI18NEJBDeployed.jar	Contains the WebSphere LocalizableText EJBs and dependent classes
ACServicesSMClient.jar ACServicesSMConsole.jar	Contains the classes for the AC System Management Console
jmxri.jar	Contains the Java Management Extensions (JMX) Instrumentation and Agent Reference Implementation from Sun Microsystems
ACModel.jar	Contains the AC deployment model classes that are required to run the AC System Management Console

Tip: To find the name of the .jar file that contains a specific class amongst multiple .jar files (for example, to figure out which .jar to put in the classpath):

1. Click **Start** and select **Find** and then select **Files or Folders**.
2. In the Name & Location panel, provide the general location in which to search (for example, *C:\Program Files\WSBC*).
3. In the Advanced panel, enter the name of the class in the **Containing Text** field.
4. Click **Find Now**.

The results of the search displays the .jar file that contains the class.

Deploy the AC Services in the VisualAge for Java WebSphere Test Environment

Configure VisualAge for Java for AC Services

In the VisualAge for Java Workbench, right-click the **IBM WebSphere Test Environment** project and select **Manage > Create Open Edition** from the pop-up menu. Repeat this step and create an open edition for the package **com.ibm.websphere.command** (you can find this package in the same project). You must perform this step because one of the Beans from the AC Services adds a generated class to this project.

Import third-party and AC Services files into projects

Create projects named after the ones specified in the following table. (You may use any other unique name for these projects.) Import the associated .jar files into these projects:

Project name	.jar file
JMXRI	jmxri.jar
WSBCACServices	ACServicesServer.jar
WSBCSystemManagementConsole	ACServicesSMConsole.jar
IBM WebSphere Localizable Text	

Note: The `ACServicesSMConsole.jar` has a dependency on the `ACModel.jar`. Refer to the Tools section in this document for more information on this file and how to import it. The `ACServicesSMConsole.jar` has a dependency on the `ACModel.jar`. In VisualAge for Java, create a project called `WSBCACModel`, and import the `ACModel.jar` from the `<WSBCROOT>\tools\lib` directory into the `WSBCACModel` project.

Note: Do not import a .jar file into the IBM WebSphere Localizable Text project. The LocalizableText EJB group will use this project later.

For example, the following steps show you how to import the `jmxri.jar` file into the project `JMXRI` (you would import the other files in a similar manner):

1. In the Workbench, click the **Projects** tab. On the menu bar, click **Selected > Add > Project**.
2. In the **Add Project** SmartGuide, in the **Create a new project named** field, enter the name of the project (`JMXRI`). Click **Finish**.
3. In the Workbench, right-click the project you have just created (`JMXRI`) and select **Import** from the pop-up window.
4. In the **Import** SmartGuide, select the **Jar file** radio button as your import source. Click **Next**.
5. In the **Filename** field, specify the full path name of the .jar file you want to import (`jmxri.jar`).
6. Select the **.class** check box. Click the **Details** button next to this check box. In the **Class file import** window, click **Select all**. Click **OK**.
7. Repeat step 6 for the **resources** check box.
8. Click **Finish**.

Import AC Services EJB .jar files into EJB groups

Create EJB groups named after the ones in the following table. (You may use any other unique name for these EJB groups.) Associate each EJB group with the specified project. Afterwards, import the specified EJB .jar file.

EJB group name	Associated project	EJB .jar file	Beans
ACServices	WSBCACServices	ACServicesEJB.jar	ACCommandTarget MBeanServerConnector
LocalizableText	IBM WebSphere Localizable Text	ACEI18NEJB.jar	LocalizableTextResourceAccessor

For example, the following steps show you how to create the ACServices EJB group and import the required Beans (you would create the other EJB groups in a similar manner):

1. In the Workbench, select the **EJB** tab. On the menu bar, click **EJB > Add > EJB Group**.
2. In the **Add EJB Group** SmartGuide, in the **Project** field, specify the name of the project that will contain the EJB group (*WSBCACServices*). In the **Create a new EJB group named** field, enter the name of the EJB group (*ACServices*). Click **Finish**.
3. Right-click the **ACServices** EJB group and select **Import Enterprise Beans** from the pop-up menu.
4. In the **Import from an EJB JAR File** SmartGuide, in the **Filename** field, specify the full path name of the .jar file you want to import (*ACServicesEJB.jar*).
5. Ensure that the check boxes for **beans**, **.class**, and **resource** are selected.
6. Click **Finish**. The EJB group should now contain all of the Beans in the .jar file.
7. Each Bean should already have some predefined deployment information, including its JNDI name, transaction attributes, and environment variables. If you want to view or change these, right-click on the Bean and select **Properties** from the pop-up menu.
8. Redeploy each Bean in the EJB group. Right-click each Bean and select **Generate Deployed Code** from the pop-up menu.
9. Modify the exclude list:
 - a. Open the `<VisualAge for Java Root>\ide\project_resources\IBM WebSphere Test Environment\exclude.list` file in a text editor.
 - b. Ensure that the following entries are in this file:
 - `com.ibm.websphere.i18n.localizabletext._LocalizableText_Stub`
 - `com.ibm.websphere.i18n.localizabletext._LocalizableTextL_Stub`
 - `com.ibm.websphere.i18n.localizabletext._LocalizableTextLTZ_Stub`
 - `com.ibm.websphere.i18n.localizabletext._LocalizableTextTZ_Stub`
 - c. Save the file if necessary.

Configure AC Services

Edit the following .ini files to configure the AC Services:

com/ibm/wsbc/acs/ACServices.ini

Find this file in the <VisualAge for Java Root>\project_resources\WSBCACServices directory. This file lets you configure AC NLS, AC Context, and AC Exception Logging Support Services.

com/ibm/wsbc/acs/sm/ui/SMConsole.ini

Find this file in the <VisualAge for Java Root>\project_resources\WSBCSystemManagementConsole directory. Change the value of CFG_developmentTime to ~~true~~ **false** if you want to ~~use the console in VisualAge for Java~~ **test the AC in VisualAge for Java in conjunction with the AC Deployment Tool.**

Run the System Management Console

To run the System Management Console in VisualAge for Java, complete the following steps.

1. Select the class SMConsoleFrame. Right-click and select **Properties**. Go to the **Classpath** tab and add the WSBCACModel project to the classpath.
2. Select and run the SMConsoleFrame class.

Deploy the AC Services in WebSphere Application Server

Add .jar files to the server's dependent classpath

1. In the tree view, select the node that is named after your server.
2. In the right-side pane, append the following string to the **Dependent classpath** field (without the line breaks or any intervening white space):

```
<WSBC directory>\<AC Services subdirectory>\ACServicesServer.jar;  
<WSBC directory>\<AC Services subdirectory>\jmxri.jar;
```

Select **Apply**.

Add .jar files to the command line argument of the server's classpath

1. Open the WebSphere Administrator's Console.
2. On the menu bar, select **View > Topology**. Expand the node named after your server. Select the application server where you want to deploy an AC (for example, **Default Server**).
3. In the right-side pane, enter the following string to the **Command line arguments** field (without the line breaks or any intervening white space except the one after "classpath"):

```
-classpath  
<WSBC directory>\<AC Services subdirectory>\ACServicesEJBDeployed.jar;  
<WSBC directory>\<AC Services subdirectory>\ACServicesServer.jar;  
<WSBC directory>\<AC Services subdirectory>\ACEI18NEJBDeployed.jar;  
<WSBC directory>\<AC Services subdirectory>\jmxri.jar;
```

Select **Apply**.

Deploy AC Services EJBs in WebSphere Application Server

For each application server where you want to deploy an AC, deploy the following Beans from their respective .jar files as listed in the table:

<i>AC Services EJBs</i>	
EJB	.jar file
LocalizableTextResourceAccessor	ACEI18NEJBDeployed.jar
ACCommandTarget, MBeanServerConnector	ACServicesEJBDeployed.jar

Note that there are two Beans stored in the ACServicesEJBDeployed.jar file. The following steps show you how to deploy these AC Services for a given application server:

1. On the menu bar, click **View > Topology**. Expand the node named after your server. Expand the application server where you want to deploy an AC (for example, **Default Server**). Right-click the container where you want to deploy an AC (for example, **Default Container**), and select **Create > EnterpriseBean** from the pop-up menu.

Note: If you are going to redeploy a Bean that already exists in a container, you first must stop, then remove that Bean. Afterwards, you may recreate that Bean.

2. In the **Create EnterpriseBean** window, select the **General** tab. In the **Jar file** field, specify the .jar file by following these steps:
 - a. Click on the **Browse** button (be prepared to wait 1-2 minutes).
 - b. Double-click the file you want to deploy. For example, to deploy the LocalizableTextResourceAccessor, you would double-click the ACEI18NEJBDeployed.jar file.
 - c. Double-click the .ser file for the Bean you want to deploy. For example, if you double-clicked the ACEI18NEJBDeployed.jar file, you would double-click the `com.ibm.websphere.i18n.localizabletext.LocalizableTextResourceAccessor/LocalizableTextResourceAccessor.ser` Bean.
 - d. You will be asked, "This jar is not enabled for Work Load Management. Would you like to enable it now?" Click **No**.

WebSphere Application Server should have automatically filled in the **Name** and **Deployment descriptor** fields for you.

Note: If you use more than one server to host multiple ACs, you must ensure that the JNDI home names of the ACCommandTarget and the MBeanServerConnector EJBs are different for each server. Therefore, you may have to change the **Deployment descriptor** fields.

The MBeanServerConnector Bean home name must be the ACCommandTarget Bean home name combined with the fixed name "MBeanServerConnector" like this (without any line breaks or intervening spaces):

```
commandTargetHomeNameMBeanServerConnector
```

The variable `commandTargetHomeName` is the home name used for the ACCommandTarget Bean in that server.

The LocalizableTextResourceAccessor Bean must also have a unique home name for each EJB server. If a new home name is assigned to LocalizableTextResourceAccessor, the corresponding entry in `ACServices.ini` must be changed, as described in "Configure the AC Services".

3. Click **OK**.
4. Right-click the Bean that you have just created in the tree and select **Start** from the pop-up menu.

Repeat steps 1-4 for the rest of the Beans.

Configure the JNDI database

A common problem you may encounter is that the AC Command Service uses JNDI to register the command properties, using as the name `<AC instance name + function name + properties>`. This often exceeds the default length for JNDI names, which is 64 characters. To use JNDI names that are longer than 64 characters, you have to alter the naming server's tables in the database to a length that can handle the JNDI name. In most circumstances, a length of 128 is sufficient. To do this for DB2, open a DB2 command line processor and type the following commands, replacing `databaseName` with the name of the database you are using (WAS for WebSphere). This example sets the maximum name length to 128.

```
CONNECT TO databaseName
```

```
ALTER TABLE EJSADMIN.BINDINGBEANTBL ALTER COLUMN NAME SET DATA  
TYPE VARCHAR (128)
```

```
ALTER TABLE EJSADMIN.PROPERTYBEANTBL ALTER COLUMN NAME SET DATA  
TYPE VARCHAR (128)
```

```
ALTER TABLE EJSADMIN.PROPERTYBEANTBL ALTER COLUMN PROPERTYNAME  
SET DATA TYPE VARCHAR (128)
```

```
DISCONNECT databaseName
```

Add .jar files to the client's classpath

Make sure that the following .jar files appear in the client's classpath:

- `ACServicesEJBClient.jar` (contains EJB classes for the client)
- `ACServicesClient.jar` (contains other classes for the client)
- `ACEI18NEJBClient.jar` (contains IBM WebSphere LocalizableText classes that are required for the client)

If the client of the AC is itself an AC, the above files are not needed. The corresponding server-side .jar file already contains the classes required for the client.

If the client of the AC is running in a stand-alone JDK, add the `ujc.jar` and `xml4j.jar` files to the client's classpath in addition to the above .jar files. You can find these files in the `C:\WebSphere\AppServer\lib` directory. The `ujc.jar` file satisfies the RMI-IIOP calls made by the EJB from the client to the server.

Add .jar files for the AC System Management Console's classpath

The AC System Management Console is a stand-alone Java application. Add the following .jar files to the client's classpath:

- `ACServicesSMConsole.jar` (contains the console classes)
- `ACServicesSMClient.jar` (contains additional console classes)

- `jmxri.jar`
- `ACModel.jar`
- `ujc.jar` (Find this file in the `C:\WebSphere\AppServer\lib` directory.)

You may run the System Management Console from a machine different from the one where you deployed the ACs.

Configure the AC Services

You may extract and customize the file `ACServices.ini` contained in the file `ACServicesServer.jar`.

If a new JNDI home name (such as `newHomeName`) was assigned to the `LocalizableTextResourceAccessor` Bean, find the following entry in the `ACServices.ini` file:

```
formatterHomeName=com/ibm/websphere/i18n/localizabletext/LocalizableTextResourceAccess  
orHome
```

Change the above entry to the following:

```
formatterHomeName=newHomeName
```

Whatever JNDI home name you use must be unique to the server.

To apply the changes you have made in this file, you may do either one of the following:

- Recreate the `ACServicesServer.jar` with your version of `ACServices.ini`
- Add your version of `ACServices.ini` *before* the file `ACServicesServer.jar` in the classpath, so that your version of the file takes precedence. You must specify the full path name of `ACServices.ini`.

Configure the System Management Console

You may extract and customize the file `SMConsole.ini` contained in the file `ACServicesSMConsole.jar`.

To apply the changes you have made in this file, you may do either one of the following:

- Recreate the `ACServicesSMConsole.jar` with your version of `SMConsole.ini`
- Add your version of `SMConsole.ini` *before* the file `ACServicesSMConsole.jar` in the classpath, so that your version of the file takes precedence. You must specify the full path name of `SMConsole.ini`.

Advanced Components

This section contains the following information:

- Install and deploy Advanced Components in WebSphere
- Install, deploy, and run Advanced Components in VisualAge for Java

Install and deploy Advanced Components in WebSphere

This section provides high-level descriptions of the steps that must be performed to install and configure one of the Advanced Components on the Version 1.1 Studio CD. While each step is only a high-level description, they also identify where to find more information should you require it. For example, information related to an Advanced Component can be found in the online information included on the Studio CD, in either the component's documentation or in the general section for Advanced Components.

Before you begin to install and deploy an AC, check that the WSBC Install Shield has installed the AC code. If the default path was used, there should be a \Program Files\IBM\WSBC\1.1\ACFeatures\Components\ directory. You must also have installed and configured the AC Deployment Tool as this tool is used in deploying the AC. You must also have installed the AC services as the ACs require most or all of these services to run. The process of installing and deploying an AC involves the following major steps:

1. Creating the database and its database tables.
2. Configuring the datasources in the application server.
3. Configuring the AC by making changes to its configuration file.
4. Running the AC Deployment Tool.
5. Deploying and registering the AC in WebSphere.

Each step is described in more detail below.

In addition, if you want to see design models of one of the ACs or one of the parameter suites, you can install the appropriate .cat file in Rational Rose. For information on how to do this, see "Install an AC .cat file in Rational Rose".

Create the database and its tables

For each AC, create the database that it uses. The following table shows the default name of the database that you should create for each AC. If you do not use the default database names, you must make appropriate changes in the bat files that create the tables for the database. If you do not use the default high-level qualifier of USERID, make the appropriate change to the ACImplementation.xml file as well.

<i>Windows NT DB2 batch files</i>		
AC	Database name	Batch file to create the columns (DB2 only)
Customer Profile	cpBoth	<pre><CD>: \Program Files\IBM\WSBC\1.1\ACFeatures\ Components\CustomerProfile\WIN\db2\ customerProfileEnterpriseDB.bat <CD>: \Program Files\IBM\WSBC\1.1\ACFeatures\ Components\CustomerProfile\WIN\db2\ customerProfilePersonDB.bat</pre>

Windows NT DB2 batch files		
AC	Database name	Batch file to create the columns (DB2 only)
Product Catalog	proCat	<CD>:\Program Files\IBM\WSBC\1.1\ACFeatures\Components\ProductCatalog\WIN\db2\ ProductCatalogACTables.bat

Version 1.1 of the AC requires a specific database structure. The Persistence section of the online AC documentation (see **<component name>Tell me about >Version**) contains a diagram that shows the tables and their relationships within the structure. You can also consult the ACImplementation.xml file to obtain the default names of the tables and their columns. If you are using DB2 for Windows NT, you can use the provided batch files to create the tables for the database using the following format and arguments:

- For Customer Profile:
`<batchfile.bat> <databaseName> [<userID> <password>] [<qualifier>]`
- For Product Catalog:
`<batchfile.bat> [<userID> <password>] [<qualifier>]`

where databaseName is the name of the database you are creating, and the optional arguments of userID, password, and qualifier are the user ID, password, and qualifier that you use to access the database.

If you are not using DB2 for Windows NT, you must create the tables manually. Consult the documentation accompanying the database product for information on how to create the tables.

Note: Although you cannot change the structure of the database, you can customize the names of the tables and columns, add columns, and set the maximum size of the columns.

Configure the datasources in the application server

This step makes the application server aware of the databases you created earlier. To do this:

1. Start the WebSphere Administrator's Console. You can do this by selecting **Programs**, then **IBM WebSphere**, then **Application Server**, and then **Administrator's Console**.
2. For each datasource, configure it by selecting **Types View** and then right-clicking **DataSource** and selecting **Create**. For the datasource name, use its name (for example, proCat for the Product Catalog AC database) and not its URL. For the JDBC Driver, use the default Admin DB Driver.
3. **Click OK. If you are configuring multiple Customer Profile datasources, perform the following steps.**
 - a. **Return to the Topology View and select the default Admin DB Driver.**
 - b. **Change the JTA Enabled field to True and click Apply.**

Configure the AC by making changes to its configuration file

This step enables you to customize the ACImplementation.xml file, which is used to create the AC instance. If, for example, you have customized the name of a column, you must change a setting within the ACImplementation.xml file for the AC.

If you are not using the default values for the database qualifier and datasource name specified in the ACImplementation.xml file, you must change the appropriate settings in that file. For example, for the Customer Profile AC, you would change the personDataSource, enterpriseDataSource, personDBQualifier, and enterpriseDBQualifier. For other customizations that you can make, see the Customize page and the Configure page for the AC.

The ACImplementation.xml file is in the main .jar file for the AC. The system management portion of the file is enclosed within the <modelMBeanInfoDescriptor> </modelMBeanInfoDescriptor> tags. At deployment time, the AC Deployment Tool stores the system management portion of the ACImplementation.xml file into its persistent storage.

Run the AC Deployment Tool

This step determines dependencies, registers the name of the AC instance, and creates the .jar files for the System Console. It involves the following steps

1. Create a copy of the original .jar file and store the copy in the deployment directory. Use the original version of the file for runtime.
2. Run the AC Deployment Tool to create the output copy of the .jar file.
3. To update the AC with any changes made using the AC Deployment Tool, copy the ACDD.xml file from the output copy of the .jar file to the runtime copy.

You can use the default settings provided by the AC. If you change the name of the AC instance or the JNDI name of the AC's Session Bean (the Function Group Registered Name as it is called by the tool), note these changes as you will need them when deploying the AC's EJBs.

For more information on the AC Deployment Tool, see the online information. The .jar files used by the tool to create the output .jar files are located in the following directories on the CD:

<i>Location of the AC .jar files</i>	
AC	.jar file directory
Customer Profile	<CD>:\Program Files\IBM\WSBC\1.1\ACFeatures\ Components\ CustomerProfile
Product Catalog	<CD>:\Program Files\IBM\WSBC\1.1\ACFeatures\ Components\ ProductCatalog

Deploy and register the AC in WebSphere

The following are high-level steps to deploy and register the AC. For more information on each step including, for example, which .jar files to add to the classpaths and which Beans to deploy on the EJB server, see "Deploying an AC" (expand **Advanced Component** and then expand **General Information**) in the online information.

1. Open the WebSphere Administrator's Console.
2. Add the .jar files to the command line argument of the server's classpath.
3. Add the .jar files to the server's dependent classpath.
4. Deploy the EJBs in the EJB server.
5. Add the .jar files to the client's classpath.

Install, deploy, and run an AC in VisualAge for Java

This section provides descriptions of the steps that must be performed to install, deploy, and run one of the Advanced Components in VisualAge for Java. The steps use the Customer Profile AC as an example, but the steps are similar for Product Catalog (except that Product Catalog does not have a dependency on the CommonCustomerSuite).

Before you begin to install, deploy, and run an AC in VisualAge for Java, check that the WSBC Install Shield has installed the AC code. If the default path was used, there should be a \Program Files\IBM\WSBC\1.1\ACFeatures\

Components\ directory. You must also have installed the AC services as the ACs require most or all of these services to run. The process of installing and deploying an AC involves the following major steps:

- Preparing the VisualAge for Java environment.
- Importing the ACs and their dependencies.
- Importing and deploying the EJBs of the AC.
- Running the AC.

In addition, if you want to see design models of one of the ACs or one of the parameter suites, you can install the appropriate .cat file in Rational Rose. For information on how to do this, see "Install an AC .cat file in Rational Rose".

Import the ACs and their dependencies

For each .jar file for the AC (and its dependencies), create a project and, in the order shown below, import the associated jar file :

Customer Profile		
Order	Project	.jar file
1	WSBCCCommonParameterSuite	CommonParameterSuite.jar
2	WSBCCCommonCustomerSuite	CommonCustomerSuite.jar
3	WSBCCCommonCoreAC	CommonCoreAC.jar
4	WSBCCustomerProfileAC	CustomerProfileAC.jar

Product Catalog		
Order	Project	.jar file
1	WSBCCCommonParameterSuite	CommonParameterSuite.jar
3	WSBCCCommonCoreAC	CommonCoreAC.jar
4	WSBCProductCatalogAC	ProductCatalogAC.jar

Note: If you are importing both ACs, you only need to import the CommonParameterSuite.jar and the CommonCoreAC.jar once.

Import and deploy the EJBs of the AC

Once you have imported the AC's .jar files and created a project for the AC, you can then import and deploy the EJBs of the AC into VisualAge for Java:

1. From the Workbench, select the **EJB** tab.
2. Create a Bean group and call it, for example, CustomerProfileAC and associate it with the appropriate project such as the WSBCCustomerProfileAC project.

3. Right-click the new Bean group and select **Import Enterprise Beans**.
4. Browse to find the EJB jar file for the AC. For example, for the Customer Profile AC, the jar file is CustomerProfileACEJB.jar.
5. Ensure that the checkboxes for importing **beans**, **.class**, and **resources** are selected.
6. Click **Finish**. The Bean group should now contain all of the Beans for this AC.
7. Each Bean should already have some predefined deployment information, including its JNDI name and AC instance identifier. If you wish to view or change these values, right-click the Bean and select **Properties**. If you changed the name of the AC instance or the JNDI name of the Session Bean when you deployed the AC using the AC Deployment Tool, you must also make these changes to the Bean's properties.
8. Redeploy each Bean in the group by right-clicking the Bean and selecting **Generate Deployed Code**.
9. Select the **EJB** tab. Select (by control-clicking) the EJB group of the AC you want to deploy, the ACServices EJB group, and the LocalizableText EJB group. Right-click one of the selected groups and this multiple selection and select **Add To** and then select **Server Configuration**. This step adds all the EJBs from the groups you selected to the same EJB server.
10. Start the Persistent Name Server through the WebSphere Test Environment.
11. In the EJB Server Configuration window, right-click the EJB Server that you have just created and select **Start Server** from the pop-up menu.
12. Start the AC System Management Console by starting the class `com.ibm.wsbc.ac.acs.sm.ui.SMConsoleFrame`.

Note: In development mode, the AC System Management Console can only support one EJB server. The AC System Management Console is in development mode if the value for `CFG_developmentTime` is set to true in the `SMConsole.ini` file. Therefore, if you have more than one AC that you want to deploy, you must add those ACs to the same EJB server.

See the topic "Manage AC instances" in the WSBC online information for more about the AC System Management Console. You can find this topic in the **Advanced Components > General Information > How do I** section of the online information.

Run the AC

Once you have imported and deployed the AC's EJBs, you are ready to run the AC. To do this, follow these steps:

1. From the Workbench, select the **Workspace** menu and then select **Tools** and then select **Launch WebSphere Test Environment**. This opens the WebSphere Test Environment Control Center.
2. If the **Persistent Name Server** is not running, start it.
3. From the Workbench, select the **EJB** tab and right-click the Bean group for the AC. Select **Open To** and then select **Server Configuration**.
4. In the EJB Server Configuration dialog, for each of the EJB servers associated with the AC and with the AC services, right-click the server and select **Start Server**.
5. Use the System Management AC service to install and start the AC. For instructions, see the topic "Manage AC instances" in the WSBC online information. You can find this topic in the **Advanced Components > General Information > How do I** section of the online information.

The AC should now be available for receiving and processing requests.

Samples and examples

For information on installing, configuring, and deploying the samples included on the Studio Version 1.1 CD, see the Samples and Examples Installation Guide ([wsbc_samples_install.pdf](#)).

Help system

This section describes how to install, run, and if needed, troubleshoot the WSBC Help System, which contains the documentation for IBM WebSphere Business Components Studio Version 1.1.

Install the Help System

When you install the WSBC product by running the **Setup.exe** file, you can install the Help System by doing either of the following:

- Select the **Typical** installation option, which automatically installs the documentation and all of the product's components.
- Select the **Custom** installation option and then select **Documentation** from the list of product components available to install.

Either option will install the NetQuestion search engine and the Help System on your local machine.

Run the Help System

You can run the WSBC Help system by doing either of the following:

- From the **Start** menu, select **IBM WebSphere Business Components 1.1** and then **Documentation**
- From the WSBC desktop icon, open the **Documentation** folder, and then click the **Documentation** icon

Configure the Studio help system to run on a Windows server

You can install and configure the WSBC Studio help system to run on a Windows NT or Windows 2000 Web server, so that other members of your organization can view and search the online help over a network without having to install the NetQuestion and help system code on their workstations.

To perform this network installation, you must have Web server software, such as Microsoft Peer Web Services (PWS) or Microsoft IIS, installed on the server. The Web server should have a scripts directory. (In PWS and IIS, your scripts directory is usually located under `x:\inetpub`.)

After installing the server software, complete the following steps:

1. Install the WSBC Studio product on the server.
2. Copy the following files from the NetQuestion installation directory (for example, `x:\imnq`) to the `x:\inetpub\scripts` directory, if they are present in the NetQuestion installation directory:

```
vahwebx.exe
vahwebx.cat
vahhelp.cfg
wsbcfoot.htm
wsbchead.htm
wsbcsrch.exe
*henus.htm
```

3. Edit the `immmap.dat` file, which is in the `netq_install_dir\instance\help\data\` directory. This file contains the starting part of the Web address for each index. Change all occurrences of the substring `http://localhost:49213/cgi-bin` to

`http://server.city.domain.organization/scripts` (such as
`http://cobweb.stl.ibm.com/scripts`).

This change causes the links generated for search hits to yield a remote (non-localhost) Web address, so that a user can follow them from any computer.

4. In the documentation installation directory (for example, `x:\ibm\wsbc\1.1\doc`), edit `hgssrch.htm` and `hgcsrch.htm`. Change all occurrences of `http://localhost:49213/cgi-bin` to `http://server.city.domain.organization/scripts`.
5. Copy `wbcstar*.gif` in the NetQuestion installation directory to a new `icons` subdirectory under `inetpub\wwwroot`.
6. Edit the product configuration file, `wsbchelp.cfg` in the documentation installation directory, and change the following entries to the values shown:

```
HTML_HOSTNAME=server.city.domain.organization
CGI_BIN_DIR=scripts
START_LITE_DAEMON=0
START_NETQ_DAEMON=0
```

You have now completed the steps required to configure the Studio help system to run on the server.

To run the Studio help system on the server, launch the following URL from any browser:

`http://Hostname/scripts/vahwebx.exe/help/wsbc/Extract/0/index.htm`

Help System Troubleshooting

This section contains solutions to problems that you may experience when using the Help System.

Help System does not launch

If nothing happens or the Help System does not display properly when you try to launch it, do the following:

1. Open a command prompt.
2. Change to the NetQuestion installation directory (for example, `x:\imnq_nt`)
3. Run the following command:

```
httpd1 -r httpd.cnf
```
4. Launch the Help System again (either via the **Documentation** desktop icon or the **Start** menu).

Help System for VisualAge for Java or DB2 does not run

If after you install the WSBC Help System, the Help System for VisualAge for Java or DB2 will not run, you will need to run the **Repair** installation option for those products, which will reinstall their Help Systems.

Other problems

If you experience other problems when using the Help System, it is recommended that you reinstall the Help System, as follows:

1. Insert the WSBC Studio CD into your CD-ROM drive.
2. Run the **Setup.exe** file from the **WSBCInstall** directory.
3. When you get to the **Modify, repair, or remove the program** screen, select **Modify**.

4. On the resulting screen, deselect **Documentation** so there is no longer a checkmark beside the option, and then click **Next**.
5. When asked whether you want to remove the NetQuestion search engine from your machine, click **No**. This prevents uninstalling NetQuestion from other IBM products that currently use it.
6. After the documentation has been successfully uninstalled, relaunch the **Setup.exe** file.
7. When you get to the **Modify, repair, or remove the program** screen, select **Modify**.
8. On the resulting screen, select **Documentation**, and then click **Next**.

Notices

IBM may not offer the products, services, or features discussed in this document in all 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.

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 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 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:

IBM Canada Ltd.,
Department 071,
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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems.

Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

Trademarks and service marks

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

AIX
CICS
DB2
DB2 Universal Database
e-business
IBM
LANDP
MQSeries
OS/2 Warp
OS/390
RS/6000
SanFrancisco
VisualAge
Visual Banker
WebSphere

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

Rational Rose is a registered trademark of Rational Software Corporation.

Other company, product, and service names may be trademarks or service marks of others.

End of document