

IBM Product Connectivity Scenarios and Patterns
Version 1 Release 0

*Scenario: Deploying a MQTT mobile
application with WebSphere MQ and
Liberty profile*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 13.

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

© **Copyright IBM Corporation 2014.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

| | | | |
|---|----------|---|-----------|
| Figures | v | Deploying the sample application to the Liberty profile | 8 |
| Chapter 1. Scenario: Deploying an MQTT mobile application with WebSphere MQ and WebSphere Application Server Liberty profile | 1 | Deploying the sample application to mobile devices | 10 |
| | | Deploying the sample application to an iOS device | 10 |
| | | Deploying the sample application to an Android device | 10 |
| | | Verifying the solution | 11 |
| Chapter 2. Planning the solution | 3 | Notices | 13 |
| Assumptions | 3 | Programming interface information | 15 |
| Business overview | 3 | Trademarks | 15 |
| User roles and interactions. | 4 | | |
| Chapter 3. Implementing the solution | 7 | Sending your comments to IBM | 17 |
| Configuring the MQTT service with WebSphere MQ | 8 | | |

Figures

| | | | |
|--|---|---|---|
| 1. User role interactions to deploy the solution | 5 | 2. Delivered logical topology | 7 |
|--|---|---|---|

Chapter 1. Scenario: Deploying an MQTT mobile application with WebSphere MQ and WebSphere Application Server Liberty profile


Mobile devices can connect to Java™ EE applications on WebSphere Application Server Liberty profile by using the MQTT protocol. Starting with an existing WebSphere MQ and Liberty profile installation, this scenario leads you through the key tasks that are required to connect a mobile device to Liberty profile.

This scenario was developed using sample installations, and uses sample applications to demonstrate the use of Liberty profile connected to WebSphere MQ. If you want to try the scenario, you can set up a copy of the sample installation as described in the scenario. You can use the sample applications that are provided with the scenario to verify your progress through each stage.

Related information:

Scenarios and Patterns information center


This scenario in IBM Knowledge Center

 Product web page

WebSphere Application Server product web page

 Library page

WebSphere Application Server library page

 Product web page

WebSphere MQ product web page

 Library page

WebSphere MQ library page

Chapter 2. Planning the solution

You can connect WebSphere Application Server Liberty profile to a mobile client through WebSphere MQ. Review the topics in this section to understand what is covered in this scenario, the reasons why a business might want to follow the scenario, and the user roles that are involved.

Assumptions

This scenario makes the following assumptions about your setup.

- You are using the Windows operating system.
- You are continuing from the Getting started with the Liberty profile scenario.
- You do not have security configured for WebSphere MQ. This scenario does not describe security configuration. However, if you do have security configured for WebSphere MQ you can still complete the scenario.
- You are using graphical user interfaces of WebSphere MQ rather than command-line equivalents.

Business overview

A company wants to extend their existing Java EE application on WebSphere Application Server Liberty profile to work with messages from mobile devices.

This business overview follows on from the one in Getting started with the Liberty profile.


The company now wants to offer a mobile solution to its clients, which must take the following things into account:

- Volume and cost of data that is transmitted, especially in machine to machine (M2M) messaging with limited data plans
- Power consumption of battery-powered devices
- Responsiveness (near-real time delivery of information)
- Reliable delivery over fragile connections
- Security and privacy
- Scalability

The MQ Telemetry Transport (MQTT) protocol is designed to meet these challenges.

The company wants its clients to always use the most up-to-date version of the application. Additionally, the application should be cross-platform to minimize development costs. A JavaScript application is ideal because the most recent version is loaded from the server each time it is opened. As demand rises the company wants to serve the application from WebSphere Application Server Liberty profile, where the existing Java EE application is already running.

Related information:

 https://www.ibm.com/developerworks/mydeveloperworks/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/tc_overview?lang=en

User roles and interactions

Roles used throughout these scenarios, with interactions between roles for scenario tasks.

Although roles to develop the solution are listed, this scenario focuses on deploying a solution that connects a mobile device to WebSphere MQ and Liberty profile. These roles are the same as the ones found in “Getting started with the Liberty profile”. For more information, see User roles and interactions.

Develop the solution

Develop the software aspects of the solution.

Software Architect

The Software Architect is responsible for dividing the required function between the components that make up a software solution. This person works with the specifications and standards used by existing IT systems, and determines where enhancements or new components must be written by the Developer.

Developer

The Developer is responsible for creating and testing the software components and linking them together. In some cases the Developer might need information from the Administrator, for example if the Developer must use an existing queue name in their software code. After the Developer finishes creating the software components, the Developer gives those components to the Administrator for deployment.

Deploy the solution

Deploy the solution for production use, by installing, configuring, and testing the components that provide the solution.

Administrator

The Administrator installs and configures the components that support the solution, in this case WebSphere MQ and Liberty profile. The business might have a different Administrator for each product, or one Administrator might perform installation and configuration for all components. If the Administrators are different they might need to exchange information. For example, in this scenario the WebSphere MQ administrator must give the name of the queue to the Liberty profile administrator. The Administrator also deploys the software components given to them by the Developer.

Test Implementer

The Test Implementer runs the tests to validate the solution and that the solution is ready for production use. For example, can the solution be started, stopped, backed up, and recovered after a system failure as well as be maintained?

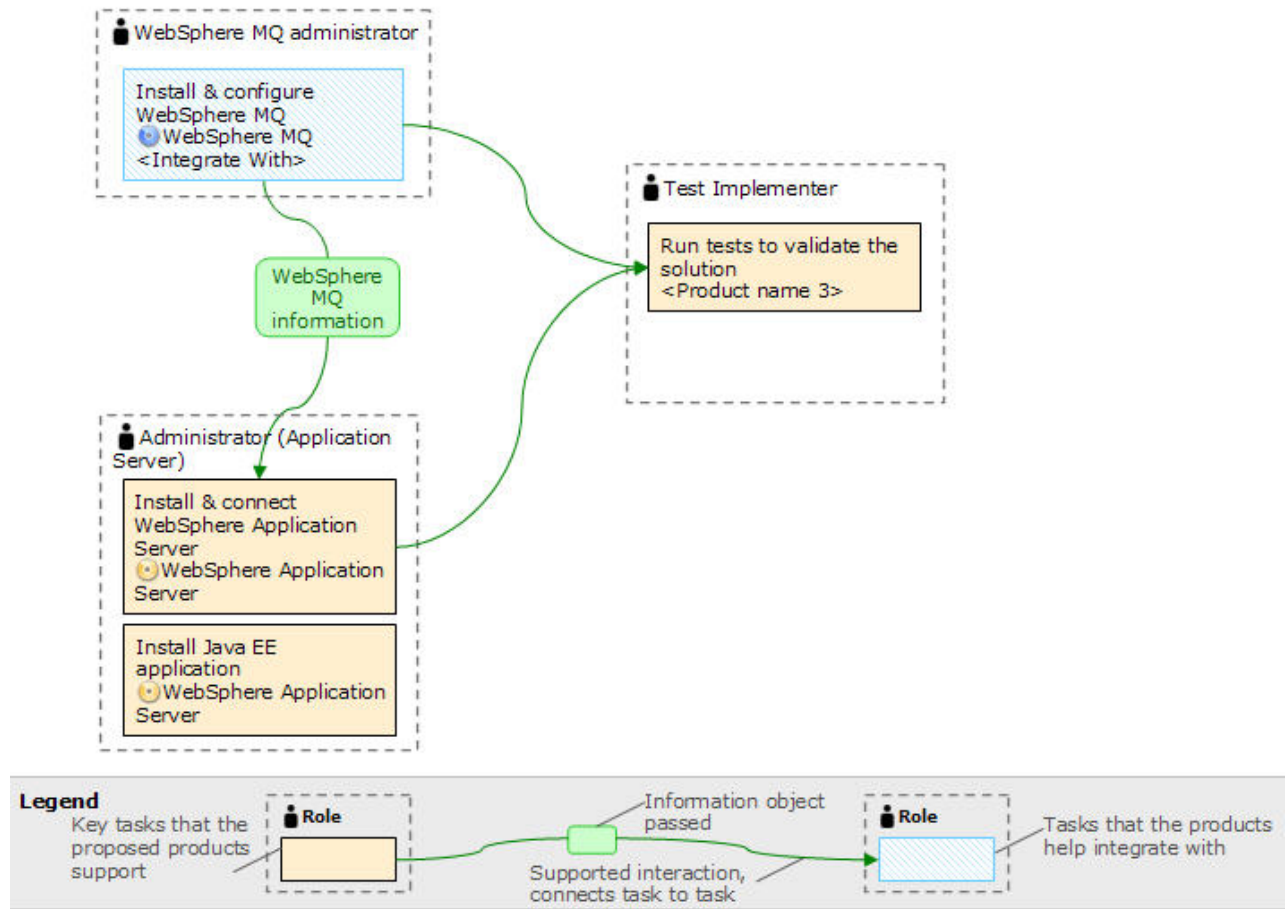


Figure 1. User role interactions to deploy the solution

Chapter 3. Implementing the solution

Implementing the solution in this scenario involves configuring WebSphere® MQ to use the MQTT protocol, deploying sample applications to WebSphere Liberty profile and mobile devices, and finally, configuring a web server.

Before you begin

The starting point for this scenario is an existing WebSphere MQ installation as the initial messaging infrastructure and an existing WebSphere Liberty profile installation like in the Getting started with the Liberty profile scenario.

About this task

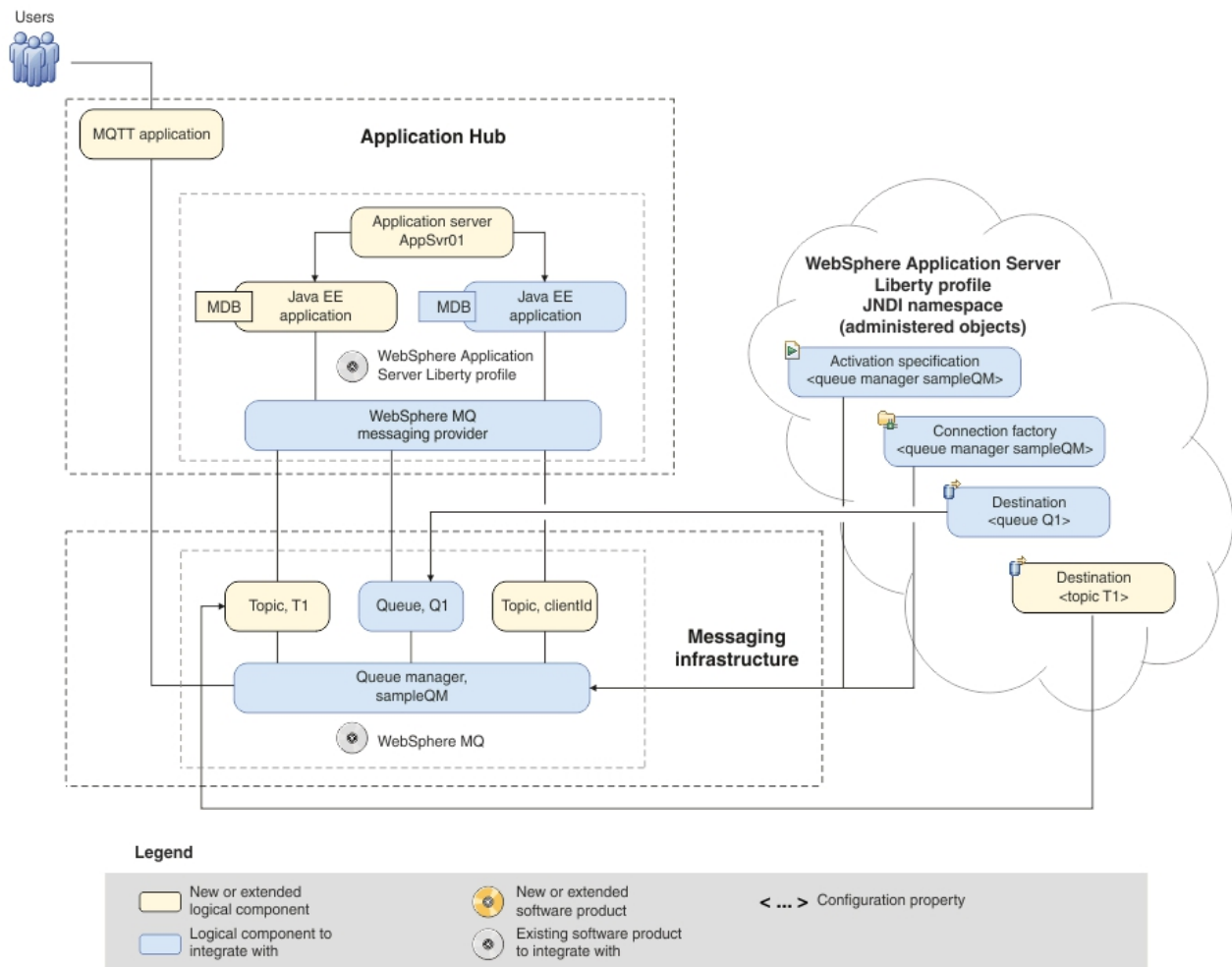


Figure 2. Delivered logical topology. The high-level logical topology diagram for new software functions delivered by integrating MQTT into the initial WebSphere MQ messaging infrastructure. This delivered logical topology shows the new functions as new or extended logical components, integrated with the software functions and products from the initial logical topology.

Configuring the MQTT service with WebSphere MQ

The MQTT service for WebSphere MQ can be installed and ready to use with the following steps.

Before you begin

- Set up and configure the sample IT configuration from *Creating a sample initial IT configuration* on your system.
- You must have local administrator authority when you are installing. Define this authority through the Windows facilities.

About this task

This task modifies an existing WebSphere MQ installation and creates an MQTT service to listen on port 1883. The configuration gives everyone permission to publish and subscribe to any topic.

Procedure

1. Modify the existing installation to include the MQTT service.
 - a. Run the same WebSphere MQ installer as in the initial sample IT configuration. This is the IBM® WebSphere MQ InstallShield Wizard. For more information, see steps 1-3 in *Installing WebSphere MQ*.
 - b. In the installer, select **Maintain or upgrade an existing instance**.
 - c. Select your existing installation and click **Next**.
 - d. Leave **Modify** selected and click **Next**.
 - e. Click the icons for **Telemetry Service** and **Telemetry Clients**, and select **Install this feature** to install each feature.
 - f. Click **Next** and then click **Modify**. The installer will stop any WebSphere MQ activity if it is running.
2. Open WebSphere MQ Explorer.
3. Run the Telemetry “Define sample configuration” wizard.
 - a. Optional: If the “MQ Explorer - Content” window is closed, click **Window > Reset Perspective**.
 - b. Expand sampleQM and select the **Telemetry** folder.
 - c. Click **Define sample configuration** to start the wizard.
 - d. Click **Finish** to create the telemetry service and run the MQTT Client Utility.
4. Test the connection in the MQTT Client Utility window.
 - a. In the Connection section, click **Connect**.
 - b. In the Subscription section, click **Subscribe**.
 - c. In the Publication section, click **Publish**. In the Client History section, you can see that a message has been published and received.

Results

You have installed the MQTT service.

Deploying the sample application to the Liberty profile

Deploy a Java EE application to the Liberty profile so that it can act as a formatter for the existing application.

Before you begin

- Set up and configure the sample IT configuration on your system, as described in Creating a sample initial IT configuration of the scenario “Connecting WebSphere Application Server Liberty profile to WebSphere MQ”.
- Download the sample applications:
 - FormatterApp.ear
 - MQTTJavascript.war
 - FormatterApp.ear
 - MQTTJavascript.war

About this task

When you set up Liberty profile in the previous scenario, you deployed the sampleMDB.ear sample application to respond to requests. The messages that are sent by the MQTT application do not contain the reply information that is required by this sample application. Therefore, you need a second application to format the incoming messages before sending them on to the sample application.

Change the server configuration by editing the wlp\usr\servers\SCENARIO\server.xml file in a text editor.

Procedure

1. Change the existing <jmsQueueConnectionFactory> tag to <jmsConnectionFactory>. The connection factory should look like the following example:

```
<jmsConnectionFactory jndiName="jms/replyCF" connectionManagerRef="ConMgr6">  
  <properties.wmqJms hostName="localhost" port="1414" channel="SYSTEM.DEF.SVRCONN"/>  
</jmsConnectionFactory>
```

2. Add a topic:

```
<jmsTopic id="myTopic">  
  <properties.wmqJms baseTopicName="T1"/>  
</jmsTopic>
```

3. Add an activation specification:

```
<jmsActivationSpec id="FormatterApp/FormatterEJB/FormatterMDB">  
  <properties.wmqJms destinationRef="myTopic" destinationType="javax.jms.Topic" port="1414"/>  
</jmsActivationSpec>
```

4. Add the application:

```
<application type="ear" id="FormatterApp" name="FormatterApp" location="FormatterApp.ear"/>
```

5. Copy the FormatterApp.ear file to the \apps folder. This is the folder where you previously put the sampleMDB.ear file. The default file path is wlp\usr\servers\servername\apps.

6. Add the web application:

```
<webApplication id="MQTTJavascript" location="MQTTJavascript.war" name="MQTTJavascript"/>
```

7. Copy the MQTTJavascript.war file to the \apps folder.

Results

The Liberty profile is now ready to accept requests from mobile devices that use the MQTT protocol.

Deploying the sample application to mobile devices

You can deploy the sample application to mobile devices. After the deployment, mobile devices can use the sample application to connect to the Java EE application.

Deploying the sample application to an iOS device

The sample application can be deployed to an iOS device.

About this task

iOS supports WebSockets by default, so you can access the web application on WebSphere Application Server Liberty profile by using the browser. Alternatively, you can make an app with a WebView to load the application without the need for a browser.

This task assumes that you access the web application on WebSphere Application Server Liberty profile by using the browser.

Procedure

1. On the iOS device, open the Safari web browser.
2. Load the web application. The URL is `http://192.168.0.101:9080/MQTTJavascript`, where 192.168.0.101 is replaced with the IP address of WebSphere Application Server Liberty profile.
3. Enter the address of your WebSphere MQ installation, and then click **Connect**. On successful connection to WebSphere MQ, the **Connect** button turns green.
4. To send a message to Liberty profile and display the response, click **Send**. If the client connects correctly in the previous step, but no response is displayed, check that Liberty profile was configured properly in “Deploying the sample application to the Liberty profile” on page 8.

Deploying the sample application to an Android device

The sample application can be deployed to an Android device.

Before you begin

Download the sample project
SampleMQTTRequestorApp.zipSampleMQTTRequestorApp.zip.

About this task

Until the default Android browser supports WebSockets, it is difficult to load the JavaScript as part of a native application like the iOS example. There are two options to consider.

1. Use an existing installation of Google Chrome or Mozilla Firefox for Android: open the browser and type in the URL of the JavaScript application.
2. Create an Android application that uses the MQTT Java libraries.

This task assumes that you use the second option.

Procedure


1. Prepare Eclipse and the Android SDK.

- a. Install a Java development environment (JDK) and an Eclipse development environment. To run the virtual device for Android, you must use the JDK from Oracle. You can obtain the JDK and Eclipse from Java SE Downloads and Eclipse IDE for Java Developers.
 - b. Add the development tools (ADT) plug-in for Android to Eclipse. For more information, see [Installing the Eclipse Plugin](#). Alternatively, the Android SDK can be downloaded which includes Eclipse and the required libraries and tools. For more information, see [Get the Android SDK](#).
2. Import the sample project.
 - a. In Eclipse, click **File > Import > Android > Existing Android Code Into Workspace**.
 - b. Click **Browse** and select the SampleMQTTRequestorApp.zip file.
 - c. Select the project in the **Project to Import** pane, select **Copy projects into workspace**, and click **Finish**.
3. Optional: Change the WebSphere MQ address in the project if it is not running on the same machine.
 - a. In the Package Explorer, navigate to: SampleMQTTRequestorApp/src/Requestor.java.
 - b. Change the address on line 17 to the correct address. The default one is 10.0.2.2:1883, which is localhost for the machine running the Android Virtual Device.
4. Configure an Android Virtual Device.
 - a. Using the Java perspective in Eclipse, click **Window > Android Virtual Device Manager**.
 - b. Click **New** and complete the fields to create a new virtual device. The project requires a target of Android 4.2.2 or above. For more information, see <http://developer.android.com/tools/devices/managing-avds.html>.
5. Run the sample application.
 - a. Right-click the project folder in the Package Explorer and click **Run As > Android Application**.
 - b. The Android emulator will take some time to start and install the application. You should be able to see an interface to request the price of beans that is similar to the JavaScript application. If the **Connected** check box is not selected, check that your WebSphere MQ server is running and configured properly in step 3.

Results

The Android device is now ready to message the Java EE application. Click the **Send** button to send a message and receive a reply from the application on the server.

Related information:

 https://www.ibm.com/developerworks/community/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/mobile_messaging_on_android

Verifying the solution

Verify that your solution works and that WebSphere MQ and Liberty profile are connected correctly.

About this task

You have now connected WebSphere Application Server Liberty profile to WebSphere MQ and installed a web application and Java EE application on Liberty profile. To verify that this has been done correctly, use the web application to send a message and display a response. You can do this through your browser, or the mobile applications provided.

Procedure

1. Test the application by using your browser.
 - a. Open your browser. The application uses the WebSocket protocol, which is supported by Mozilla Firefox 6.0 and later, Google Chrome 14.0 and later, and Safari 6.0 and later.
 - b. Load the web application. If you are on the same machine where Liberty profile is running, then this is `http://localhost:9080/MQTTJavascript` by default.
 - c. If the address of your WebSphere MQ installation is not the one displayed, enter the details and click **Connect**. The **Connect** button turns green upon a successful connection to WebSphere MQ.
 - d. Click **Send** to send a message to Liberty profile and display the response. If the client connects correctly in the previous step, but no response is displayed, check that Liberty profile was configured properly in “Deploying the sample application to the Liberty profile” on page 8.
2. Optional: Test the application by using the mobile applications and instructions provided in “Deploying the sample application to an Android device” on page 10 and “Deploying the sample application to an iOS device” on page 10.

Notices

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 grant 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 106-0032, 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 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:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

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 information 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 measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples 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 illustrate 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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software for use with this program.

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.

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information (www.ibm.com/legal/copytrade.shtml).



Sending your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

- Send an email to ibmkc@us.ibm.com
- Use the form on the web here: www.ibm.com/software/data/rcf/

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number
- The topic and page number related to your comment
- The text of your comment

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

Thank you for your participation.



Printed in USA