

WebSphere MQ Everyplace



Read Me First

Version 2.0

First Edition (November 2002)

This edition applies to WebSphere™ MQ Everyplace Version 2.0 (Program number: 5724-C77) and to all subsequent releases and modifications until otherwise indicated in new editions.

This document is continually being updated with new and improved information. For the latest edition, please see the WebSphere MQ family library Web page at <http://www.ibm.com/software/mqseries/library/>.

© Copyright International Business Machines Corporation 2000, 2002. All rights reserved.

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

About this book

This document contains the installation instructions for WebSphere MQ Everyplace Version 2.0, and other important information that you should read before trying to use this toolkit.

License warning

WebSphere MQ Everyplace is a toolkit that enables users to write WebSphere MQ Everyplace applications and to create an environment in which to run them. Before deploying this product, or applications that use it, please make sure that you have the necessary licenses.

1. The pricing of licenses for use of the Program on servers is based on 'Processor License Units'. Use of each copy of the Program on a server requires one Processor License Unit to be acquired for each processor or symmetric multiprocessor contained in the server on which the copy of the Program is to run. Different types of Processor License Units and 'Device Use Authorisations' are required, depending on whether the Program is running on point-of-sale, that is retail, equipment or on another type of computer. Use of the Program on retail equipment requires a 'Retail' server license, whereas use on other (non-retail) equipment requires a 'Network' server license.
2. Additional 'Device Use Authorisation' is required for any use of the Program on a separate client device, except those included in the Network Server license, as described at 3) below.
3. Each 'Network' server license includes authorisation for the restricted use of the Program with no more than one hundred (100) client devices, on condition that all such copies are used in the same economic enterprise or organisation as the server copy.

Device platform use authorizations, which are recorded on Proof of Entitlement documents and valid to support the use of WebSphere MQ Everyplace, are required to use the product (other than for purposes of code development and test) on specified client platforms. These licenses do not entitle the user to use the WebSphere MQ Everyplace Bridge, or to run on the server platforms specified in the WebSphere MQ Everyplace pricing group lists published by IBM® and also available on the Web via the URL mentioned below.

Please refer to <http://www.ibm.com/software/mqseries/> for details of these restrictions.

Summary of changes

This section describes changes to this edition of *WebSphere MQ Everyplace Read Me First*. Changes since the previous edition of the book are marked by vertical lines to the left of the changes.

Changes for this edition (GC34-6276-00)

In addition to editorial corrections and improvements to the text, the following information has been modified or added:

- Product version number
- Migration information
- The version 1.2.7 Programming Guide has been replaced with:
 - The WebSphere MQ Everyplace Application Programming Guide.
 - The WebSphere MQ Everyplace System Programming Guide.
 - The WebSphere MQ Everyplace Configuration Guide.
- The version 1.2.7 Native Client Information manual is now called the WebSphere MQ Everyplace C Programming Guide for Palm OS.

What is new in Version 2.0

The following section details new features and migration information for WebSphere MQ Everyplace version 2.0.

Changes

The following sections give information on new features in WebSphere MQ Everyplace.

Listener architecture

New listener classes have been created that allow the listeners to be managed by WebSphere MQ Everyplace. This means that they can be administered remotely, administered locally, and are persistent in the registry. You no longer have direct access to the implementation classes, but deal with listeners indirectly.

The old listener classes have been deprecated and moved to the deprecated.jar file. The old listeners still work, we strongly discourage you from using them.

Persistence of listeners

Listeners are now persistent in the registry. Once created they exist until removed, even if the queue manager is closed and reopened. This is different from the behaviour in version 1.2.7 and lower versions. If a listener is loaded from the registry at queue manager start-up, the listener starts automatically.

If a new listener is created then it must be explicitly started. All listeners require a name, and this name must be unique within the set of listeners on the same queue manager. This name has no purpose except to distinguish between listeners during administration and during storage in the registry.

Remote administration of listeners

Listeners can be administered by using the new admin message class MQEListenerAdminMessage. All of the local administration actions are available with the exception of check channel timeout.

Message store failure behavior

The behavior of message stores has been changed to throw MQEMessageStoreException when it encounters an irrecoverable situation in the underlying storage media, for example, if a file does not exist when it should, or a file cannot be created or destroyed. Once an MQEMessageStoreException has been thrown, the message store, and hence the queue, cannot be used until it is restarted. However, this should be a rare occurrence, but when it occurs it is serious enough to warrant manual intervention.

Registry

Improvements to the persistent data representation in the registry and the message store means changes in the format of the data. These changes mean that a registry generated under version 1 of WebSphere MQ Everyplace is not acceptable to a queue manager started using Version 2.0 of WebSphere MQ Everyplace. The application generates an exception error to signal this fact. In Java this is `com.ibm.mqe.MQEExceptionCodes.Except_QMgr_RegistryDataVersion`. If you encounter this exception you will need to migrate the data in the registry. WebSphere MQ Everyplace provides classes to perform this migration. The required classes are in the package `com.ibm.mqe.validation`, and the only public class is `com.ibm.mqe.validation.MQeDiskImageValidator`. An example of how to use this class to perform migration is provided in `examples.migration.Migrator`.

Enhancements

WebSphere MQ Everyplace version 2.0 includes many new features. It now supports even more platforms including Windows XP. There are more programming options available through the introduction of a new native C language implementation for Microsoft Pocket PC. It also supports point-to-point messaging using the Java Message Service (JMS), an integral part of Java 2 Platform, Enterprise Edition (J2EE) technology that improves programmer productivity by defining a common set of messaging concepts and programming strategies.

A subset of WebSphere MQ Everyplace now also runs on the Java 2 Platform, Micro Edition (J2ME) technology, which covers a range of smaller devices such as smart cards, pagers and small PDAs. These and many other enhancements in WebSphere MQ Everyplace 2.0 are listed below:

- Native C language implementation for Microsoft Pocket PC.
- Point-to-point messaging using a subset of the Java Message Service (JMS).
- Subset of WebSphere MQ Everyplace runs on the Java 2 Platform, Micro Edition, both CLDC/MIDP and CDC/Foundation.
- Additional platform support, including Windows XP.
- Full accessibility and improved codepage support.
- Improved trace mechanism, please see Appendix C of Application Programming Guide.

- Improved documentation.
- A new UDP Adapter, MQeUdpipBasicAdapter this replaces the MQeUdpipAdapter, this offers improved performance characteristics for high latency or unreliable networks.
- Communications listeners are now manageable using standard administration techniques.
- Scalability and quality improvements.
- Channel improvements.

If you are migrating from earlier versions of WebSphere MQ Everyplace we strongly recommend that you read the migration section of this book and the WebSphere MQ Everyplace Application Programming Guide for details of changes.

Chapter 1. Installation procedures

These procedures enable you to install WebSphere MQ Everyplace on machines that are to be used to develop WebSphere MQ Everyplace applications. In this release, the application and solution provider is responsible for deploying WebSphere MQ Everyplace to pervasive devices.

The WebSphere MQ Everyplace installation program is a Java™ jar file that has platform specific launchers that can be run straight from the product CD. The installation program extracts the working files to a temporary directory, copies the WebSphere MQ Everyplace files onto your computer, and cleans up the working files.

Migration notes

This section contains information that you might need to consider when migrating from one version or release of WebSphere MQ Everyplace to a higher version or release.

Migration information for the C codebase

For specific migration information for C codebase implemenations of WebSphere MQ Everyplace, refer to the migration information section of the WebSphere MQ Everyplace Application Programming Guide.

Migrating from version 1.2.7 to version 2.0

If you are upgrading to version 2.0 you must consider how the following changes affect your WebSphere MQ Everyplace application.

Changes to MQeFields

In order to comply with Java 2 Platform Micro Edition's (J2ME) Connected Limited Device Configuration (CLDC)/Mobile Information Device Protocol (MIDP) specification several methods have been modified or removed from MQeFields:

- The explicit use of the floating point types, float and double, have had to be removed, for example you putFloat("Val1", -1.234). Under java platforms that enable the use of float/double this functionality can be mimicked by explicitly converting the data into the equivalent int or long using the base types Java Object convert method i.e. the above method is replaced with putFloatAsInt("Val1", Float.floatToIntBits(-1.234)).

Note: Version 1 applications can retrieve these values as normal.

- Methods dumpToFile/restoreFromFile have been removed. Applications that used these functions now have to dump the MQeFields object and write the byte array to the specified file.
- Xor'ing of dumped data has also been removed due to changes made in the 'C' code base.

Peer channels

Peer channels have been removed from the WebSphere MQ Everyplace version 2.0 codebase, because they caused problems with message routing and assured delivery.

Changes to MQeChannel

The `com.ibm.mqe.MQeChannel` class has been moved and is now known as `com.ibm.mqe.communications.MQeChannel`. Any references to the old class name in administration messages is replaced automatically with the new class name.

MQeAttribute

The following changes have been implemented in relation to `MQeAttribute`:

- The implementation of the `equals()` method on `MQeAttribute` and its subclasses in version 1.2.7, and lower versions, has been renamed as `isAcceptable()`.
- An `MQeAttributeRule` now ships with the product. You should now extend your attribute rules from this class instead of `MQeRule`. All methods on `MQeAttribute` and its subclasses, which used to take an `MQeRule` object as one of its parameters now take an `MQeAttributeRule` object instead.

MQeQueueManager

In version 1 code, the `MQeFields` structure passed to the `MQeQueueManager` allowed the specification of the following two aliases. They specified the default class names to use when loading attribute keys, where an attribute key class was not specified.

- `(ascii)AttributeKey_2=com.ibm.mqe.attributes.MQeSharedKey`
- `(ascii)AttributeKey_1=com.ibm.mqe.MQeKey`

These values are hard-coded in the version 2.0 codebase, and cannot be changed using the alias mechanism. If the values are specified in .ini files, or calls to the `MQeQueueManager`, they are not used, but cause no problems.

Deprecated methods and classes

The following have been deprecated in version 2.0

- The **`MQeQueueManager.waitForMessage()`** method is now deprecated. A demonstration of an alternative way of writing a message waiting routine is provided in the `examples.queuemanager.MessageWaiter` example code.

A number of classes have been removed from the product. We recommend that you update any applications written to make use of the classes listed below to use the equivalent function provided in WebSphere MQ Everyplace version 2.0. To enable existing applications to be run during this migration, WebSphere MQ Everyplace provides the `MQeDeprecated.jar` jar file.

The `MQeDeprecated.jar` file contains the following classes:

- `MQeMQBridge.class`
- `MQeChannelListener.class`
- `MQeChannelListenerTimer.class`
- `MQeChannelManager.class`
- `MQeTraceInterface.class`

For more details on replacements for the above classes, refer to the the listing for each class in the WebSphere MQ Everyplace Java Programming Reference.

Security

The following changes have been made to security:

- The MQeCL and MQeRandom classes have been replaced with cryptoLite's CL class, supplied by IBM Zurich
- The old style mini-certificate support has been withdrawn from version 2.0

These changes have the following implications:

- The CL class will be shipped as a cryptoLite.zip. In order to use WebSphere MQ Everyplace security, the zip file must be placed in the Java class path.
- MQeMiniCertificateServer no longer supports the old style mini certificate.

You must also refer to the migration section, if one exists, of each manual for more information on upgrading to WebSphere MQ Everyplace Version 2.0.

Migrating from version 1.2.6 or lower to release 2.0

If you are migrating from version 1.2.6 or lower to WebSphere MQ Everyplace version 2.0, consider the following product changes on your existing application.

Aliases

Several Java class aliases, which were commonly used in version 1.2.6 are defaulted in version 2.0. The WebSphere MQ Everyplace Java Programming Reference section on MQeAlias() methods provides more information on this.

Attribute keys

In WebSphere MQ Everyplace version 1 code, the MQeFields structure passed to the MQeQueueManager specifies the default class names to use when loading attribute keys, where an attribute key class was not specified:

```
(ascii)AttributeKey_2=com.ibm.mqe.attributes.MQeSharedKey  
(ascii)AttributeKey_1=com.ibm.mqe.MQeKey
```

In version 2.0, you cannot change these values using the alias mechanism, because they are hardcoded. If you specify the values in .ini files, or calls to the MQeQueueManager, they are not used, but cause no problems.

MQBridge programming

In previous versions of WebSphere MQ Everyplace you had to instantiate an MQBridges object in order to connect to a WebSphere MQ system. In version 2.0, the bridge function is automatically available providing the relevant services are available on the class path. Chapter 9, "Interoperability with other messaging systems", of the WebSphere MQ Everyplace Application Programming Guide provides more information on the bridge function.

Mini-certificate server

In previous versions of WebSphere MQ Everyplace the product code included a WLTS certificate issuance server, with application programming interfaces to allow programmatic control of the server. Version 2.0 replaces this with SupportPac, ESO3, "WebSphere MQ Everyplace WTLS Mini-Certificate Server", which you can download

for free from the WebSphere MQ Web site at
<http://www.ibm.com/software/mqseries/txppacs>.

Registry

Improvements to the persistent data representation in the registry and the message store means changes in the format of the data. These changes mean that a registry generated under version 1 of WebSphere MQ Everyplace is not acceptable to a queue manager started using version 2.0 of WebSphere MQ Everyplace. The application generates an exception error to signal this fact. In Java this is `com.ibm.mqe.MQExceptionCodes.Except_QMgr_RegistryDataVersion`. If you encounter this exception you will need to migrate the data in the registry. WebSphere MQ Everyplace provides classes to perform this migration. The required classes are in the package `com.ibm.mqe.validation`, and the only public class is `com.ibm.mqe.validation.MQeDiskImageValidator`. An example of how to use this class to perform migration is provided in `examples.migration.Migrator`.

Trace

The tracing mechanism for WebSphere MQ Everyplace Version 2.0 differs from the mechanism provided by version 1 of the product. These differences are detailed in the "Migration notes" section of the WebSphere MQ Everyplace System Programming Guide.

Migrating from version 1.2.4 and 1.2.5

If you are upgrading to version 2.0 from version 1.2.4, and you use home-server and store-and-forward queues, you should be aware of the following:

- Version 1.2.4 or lower home-server and store-and-forward queues cannot talk to release 2.0 queues
- Version 1.2.5 home-server and store-and-forward queues can talk to both version 1.2.4 and lower queues, and release 2.0 queues

You need to make sure that any WebSphere MQ Everyplace systems with which your upgraded system needs to interact are upgraded to at least version 1.2.5.

Software environments

This section describes the software that you need to have installed to be able to create and run WebSphere MQ Everyplace applications.

Supported platforms

You can install WebSphere MQ Everyplace on certain server platforms only. To transfer programs and Java classes to other platforms, you must use an appropriate download or file transfer program (not supplied).

Note: You can install the C Bindings, the Native C Client, and Palm support only on platforms marked * in the following lists.

Directly supported platforms with installation support

You can install the product using the built-in tools on the following supported platforms:

- Windows NT® v4 *
- Windows® 2000 *
- Windows 95/98/ME *
- AIX Version 4.3
- Sun Solaris Version 7 and 8
- Linux Intel Kernel 2.2 (installed using a zip file)
- Linux Redhat 6.2 (Kernel 2.2)
- HP-UX 11.0

Directly supported platforms without installation support

The following platforms are supported for the testing and deployment of WebSphere MQ Everyplace, but only support installation by file transfer from another platform:

- WinCE 2.1 running on HP Jornada devices (Models 680 or 820)
- EPOC 32 bit Release 5 running on Psion devices (5MX Pro or NetBook)
- PalmOS, V3.0 or higher running on Palm V and IBM Workpad C3
- IBM 4690 OS with Java
- Pocket PCs

Indirectly supported platforms

You can use the following platforms, but IBM will only investigate problems if the problem can be reproduced on one of the directly supported platforms listed above:

- Linux on zSeries® running Kernel 2.2
- iSeries®
- OS/2
- EPOC (on devices other than those listed above)
- WinCE (on devices other than those listed above)
- QNX Neutrino
- PalmOS (on devices other than those listed above)
- Any other platform running one of the Java environments listed in “Java environment”

Java environment

Running the Java APIs requires one of the following Java runtime environments:

- IBM Java runtime (JVM V1.1 or later)
- Any Java which is Sun Java (V1.1 or later) certified
- IBM VisualAge Micro Edition *
- Personal Java *
- J2ME, refer to the J2ME section below

I

* These items are discussed in more detail below.

WebSphere MQ bridge operation requires WebSphere MQ Everyplace Classes for Java. You can download the WebSphere MQ Everyplace Classes for Java as SupportPac ma88 from the IBM WebSphere MQ Web site at <http://www.ibm.com/software/mqseries/everyplace/>. You should check the level of Java that is required to run the version of MQSeries Classes for Java that you are using.

PersonalJava

You can use PersonalJava instead of other Java runtimes on device platforms.

Using WebSphere MQ Everyplace requires the following optional classes of PersonalJava:

- To use WebSphere MQ Everyplace base classes:
 - java.io.FileInputStream
 - java.io.FileOutputStream
 - java.io.File
- To use the MQeGZIPCompressor:
 - java.util.zip.GZIPOutputStream
- To use any encryption:
 - java.math.BigInteger

The WebSphere MQ Everyplace examples require some of the optional classes in packages java.io and java.awt.

J2ME

WebSphere MQ Everyplace is compliant with the J2ME technologies, Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC), or more explicitly their most popular profiles, CDC/Foundation and CLDC/Midp.

Using CDC/Foundation enables the full functionality of WebSphere MQ Everyplace, excluding any example code requiring the awt gui package. The use of CLDC/Midp, however, restricts the application to solely 'client' side behaviour and a limited range of built-in compressors and encryptors.

Note: Devices that are CLDC/Midp enabled might have severe memory limitations that can preclude the direct use of the WebSphere MQ Everyplace core package.

WebSphere Studio Device Developer

WebSphere Studio Device Developer (WSDD) provides a Java IDE together with runtime environments for many different device platforms. WSDD includes WebSphere Micro Environment for developing J2ME applications and supports WebSphere Custom Environment for developing applications for realtime control systems and other devices in closed systems.

You can develop applications using WebSphere MQ Everyplace in the Websphere Studio Device Developer .IDE and deploy those applications using the runtime environments. WebSphere MQ Everyplace requires the use of the jclMidp class library, or higher.

C Bindings Environment

The C Bindings APIs require:

- IBM Java runtime (JVM V1.2.2 or later)
- Any Java which is Sun Java (V1.2.2 or later) certified

The C Bindings cannot be used with Personal Java or VisualAge Micro Edition.

Storage requirements

The following table shows the storage you need to perform the installation of all the available options of WebSphere MQ Everyplace:

Table 1. Storage required to perform installation

| Operating system | Storage required |
|---------------------------------|------------------|
| Windows NT (file system = NTFS) | 40Mb |
| AIX | 27Mb |
| Solaris | 27Mb |
| Linux | 27Mb |
| HP-UX | 27Mb |

Installation steps on Windows, AIX, Linux, and Solaris

Note: We strongly recommend that you uninstall any previous versions of WebSphere MQ Everyplace before installing or reinstalling this new version (see “Uninstalling WebSphere MQ Everyplace” on page 14). The installation program does not detect versions of the product prior to 1.2.4, and does not display any warnings.

At any time during the installation, click the **Back** button on a screen to take you back to previous screens and review or change information. To exit the install procedure and cancel the installation, click the **Cancel** button on any screen.

Notes:

1. On Windows your user id must have administrator access, otherwise the Start Menu icons for IBM WebSphere MQ Everyplace may not appear.
2. On AIX® you must be the root user to run the install successfully.

To install WebSphere MQ Everyplace:

1. Insert the product CD into your CD-ROM drive.
2. Start the installation in one of the following ways:
 - a. From the platform specific launcher.

The launchers are held in the platform named subdirectories on the product CD. To begin the installation process, run the correct launcher for your platform.

Notes:

- 1) If you copy the launcher to your local target machine, the setup.jar file must be copied into the parent directory of the directory in to which the launcher was copied. The launcher cannot be run from a root directory.
 - 2) If you run the install from a launcher and you see a message box with the text "No matching JVM was found" the installer was unable to find a Java environment to use. If you see this message you should use the setup.jar file for installation.
 - 3) To tell the launcher to use a specific JVM, use the following flag:
`setup.exe -is:javahome c:\jdk1.3`
 - 4) To tell the launcher to use a specific directory to store temporary files, use the following launcher flag:
`setup.exe -is:tempdir c:\mytempdir`
- b. From the setup.jar file :
- Change to the product CD directory where the setup.jar is stored and run the installation program using the Java command on your computer. This command is typically java, jre, or jview. For example:

On Windows

```
set classpath=.\setup.jar;%classpath%
jview run
```

(If you use JVM 1.2.2 or higher, you can execute the jar file by double-clicking it.)

On Linux, AIX, Solaris, and HP-UX

```
CLASSPATH=./setup.jar:$CLASSPATH
export CLASSPATH
java run
```

3. On the Welcome screen, confirm that you want to install the WebSphere MQ Everyplace program by clicking the **Next** button and then follow the prompts to complete your installation.

Note: If you wish to reinstall the product we strongly recommend that you remove any existing installations (see "Uninstalling WebSphere MQ Everyplace" on page 14).

Silent installation

You can also run the installer in silent mode. This means that no panels are displayed during the install and there are no prompts for input. As when running the install in graphical mode, there are two ways to run the install silently.

From the jar file

Run the install in the same way as previously described, but append the `-silent` flag. For example:

On Windows

```
set classpath=.\setup.jar;%classpath%
jview run -silent
```

On Linux, AIX, Solaris, and HP-UX

```
CLASSPATH=./setup.jar:$CLASSPATH
export CLASSPATH
java run -silent
```

From a platform specific launcher

Add the `-is:silent` `-silent` flags. For example:

On windows

```
setup.exe -is:silent -silent
```

On AIX

```
Setupaix.bin -is:silent -silent
```

On Linux

```
Setuplinux.bin -is:silent -silent
```

On Solaris

```
Setupsolaris.bin -is:silent -silent
```

On HP-UX

```
Setuphp-ux.bin -is:silent -silent
```

Note: If you have not uninstalled old versions of WebSphere MQ Everyplace, a silent install will use your current directory and not the default shown in the following table.

By default the installer installs WebSphere MQ Everyplace in the following directories on each platform:

| Platform | Default installation directory |
|----------|--------------------------------|
| AIX | /opt/MQe |
| Linux | /opt/MQe |
| Solaris | /opt/MQe |
| Win32 | \Program Files\MQe |
| HP-UX | /opt/MQe |

To set a different installation directory for a silent install, append the `-P MQe.installLocation` flag to the install command as follows:

From the jar file

```
java run -silent -P MQe.installLocation="C:\my new install directory"
```

From a platform specific launcher

```
setup.exe -is:silent -silent -P MQe.installLocation=
"C:\my new install directory"
```

Note: When using the `-P MQe.installLocation` parameter to override the install location during silent install, ensure that you specify the full path of the the new destination.

Using options files with silent installs

When running the install silently you can specify an options file. The options file allows you to:

- Set the install to silent
- Change the install location
- Select which features to install

The following example options file sets the install to run silently, sets the install location to "C:\MQe", and chooses to install all features except the Palm feature.

```
#specify silent install
-silent
#set features to active
-P Java.active=true
-P Documentation.active=true
-P CBindings.active=true
-P Native.active=true
-P Palm.active=false
#Set the install location
-P MQe.installLocation="C:\MQe"
```

Notes:

1. Include the "-is:silent" flag in the options file if running the install from a launcher.
2. Do not leave any blank lines in the options.txt file.
3. Start all lines with "#" or a valid command.
4. You can have multiple commands on a single line.

The following examples show how to run the installer with an options file:

From the jar file

```
java -cp setup.jar run -options C:\options.txt
```

From a launcher

```
setup.exe -options C:\options.txt
```

Alternative installation steps

The WebSphere MQ Everyplace classes are also provided as a zip file. You can use this file to install WebSphere MQ Everyplace on devices where the graphical installer is not suitable or not supported. On a UNIX[®] based system (such as Linux and HP-UX) you need to create a folder, copy the appropriate zip file into it, and then use an unzip utility to extract the class files. For example:

```
mkdir mqe
cp /cdrom/unixinst.zip mqe
cd mqe
unzip unixinst.zip
chmod -R +x *
```

Once the class files have been extracted, configure your environment to run WebSphere MQ Everyplace programs. Please see the *WebSphere MQ Everyplace Application Programming Guide* for more information.

Installed components

You can select various features during the installation of WebSphere MQ Everyplace. These features are shown below, along with the components that are installed if the feature is selected.

WebSphere MQ Everyplace for Java

WebSphere MQ Everyplace Java classes

A set of classes that implement all of the WebSphere MQ Everyplace function. Subsets of these classes can be used to provide different WebSphere MQ Everyplace configurations such as a subset for a device , or a subset for a server.

Helper classes

A set of classes derived from the base classes that implement some commonly used functions.

Example classes

A set of classes that demonstrate how to use many of the features of WebSphere MQ Everyplace. The source code for these classes is also provided.

Utilities

Tools to assist with the programming and administration of WebSphere MQ Everyplace.

WebSphere MQ Everyplace C Bindings

C Bindings specific classes

Includes the package com.ibm.mqe.bindings. These are only required for the C Bindings and do not affect existing functionality.

Header and binary files

CBindings adds in extra binary files, and header files. These files are fully documented in the *WebSphere MQ Everyplace C Bindings Programming Guide*

WebSphere MQ Everyplace C Bindings Programming Guide SC34-6280-00

This book contains guidance and procedural information for writing WebSphere MQ Everyplace C applications and administering your systems. The filename is hmq9a1_WMQE_C_BindingsProgrammingGuide.pdf.

WebSphere MQ Everyplace C Bindings Programming Reference

This book contains examples and information on structures and methods writing WebSphere MQ Everyplace C bindings applications. The filename is hmq9a1_WMQE_C_BindingsProgrammingGuide.pdf.

WebSphere MQ Everyplace for Palm OS

WebSphere MQ Everyplace for Palm Device Code

This is the Palm SupportPac™ (eap1.zip). The installer unzips and installs the code.

WebSphere MQ Everyplace C Programming Guide for Palm OS SC34-6281-00

This book contains guidance and reference information for using WebSphere MQ Everyplace on the Palm operating system. The filename is hmq8an_WMQE_C_ProgrammingGuideforPalmOS.

WebSphere MQ Everyplace for Native Platforms

Header files

The set of header files required are common for all native platforms are also shared with C Bindings.

Binary files

There are DLL and LIB files, builld against the following SDKs:

- PocketPC SDK, ARM processor and PocketPC Emulator
- PocketPC 2002 SDK, ARM processor and PocketPC 2002 Emulator

Examples

Some examples are also documented in the WebSphere MQ Everyplace C Programming Guide for Palm OS and in the WebSphere MQ Everyplace Native Platforms Programming Reference.

Documentation

English language versions of the following books are provided in Adobe Acrobat readable format:

WebSphere MQ Everyplace Read Me First SC34-6276-00

This book contains general migration and installation information for WebSphere MQ Everyplace. The filename is hmq8aa_WMQE_ReadmeFirst.pdf.

WebSphere MQ Everyplace Introduction SC34-6277-00

This book provides a general introduction to WebSphere MQ Everyplace, covering the product concepts and the relationship between WebSphere MQ Everyplace and other WebSphere MQ products. The filename is hmq8ac_WMQE_Introduction.pdf.

WebSphere MQ Everyplace Application Programming Guide SC34-6278-00

This book contains guidance and procedural information for writing WebSphere

MQ Everyplace Java and C applications and administering your systems. The filename is hmq8a1_WMQE_ApplicationProgrammingGuide.pdf.

WebSphere MQ Everyplace System Programming Guide SC34-6274-00

This book contains guidance and procedural information for writing WebSphere MQ Everyplace Java and C applications, and administering your systems. The filename is hmq8as_WMQE_SystemProgrammingGuide.pdf.

WebSphere MQ Everyplace Java Programming Reference SC34-6279-00

This book contains Java examples, and information on fields, method, and exceptionality information for writing WebSphere MQ Everyplace Java applications. This is available as a html file on the WebSphere MQ Everyplace product CD and download web site.

WebSphere MQ Everyplace C Programming Reference

This book contains C examples and information on structures, methods, and reason and return codes for writing WebSphere MQ Everyplace C applications. This is available as a html file on the WebSphere MQ Everyplace product CD and download web site.

WebSphere MQ Everyplace Configuration Guide SC34-6283-00

This book contains detailed descriptions of the WebSphere MQ Everyplace Java and C Application Programming Interface (API). The filename is hmq8ag_WMQE_ConfigurationGuide.pdf.

Notes:

1. The latest versions of these documents are available from the book section of the WebSphere MQ library Web site at: <http://www.ibm.com/software/mqseries/library/>.
2. If you need to obtain the Adobe Acrobat Reader, or would like up-to-date information about the platforms on which the Acrobat Reader is supported, visit the Adobe Systems Inc. Web site at: <http://www.adobe.com/>

Documentation in HTML

The product CD also supplies the WebSphere MQ Everyplace documentation in HTML format. These files are not included in the product installation.

To view the HTML files, open the index.html file from the html subdirectory on the product CD. This opens a page that contains links to all the WebSphere MQ Everyplace documents.

Note: You can access the latest versions of these documents from the book section of the WebSphere MQ library Web site at:
<http://www.ibm.com/software/mqseries/library/>.

WebSphere MQ Everyplace components on the Web

The product CD does not supply the following WebSphere MQ Everyplace components. However, these components are available as SupportPacs that can be freely downloaded from the WebSphere MQ Everyplace Web site at <http://www.ibm.com/software/mqseries/everyplace/>. These are essential supplements to the licensed product and include:

EAP1: WebSphere MQ Everyplace - Device code for the Palm OS

C programming language support for WebSphere MQ Everyplace application development on the Palm OS

Note: This code is also included on the WebSphere MQ Everyplace product CD.

EP0C: WebSphere MQ Everyplace - GetStarted

A tool that uses a postcard application to introduce you to WebSphere MQ Everyplace

EP01: WebSphere MQ Everyplace - Performance Report

Analyses WebSphere MQ Everyplace performance on a variety of client platforms

ES01: WebSphere MQ Everyplace - Administration Tool (MQeExplorer v1.0)

A generic tool for all Java platforms that enables easy graphical administration of WebSphere MQ Everyplace queue managers

ES02: WebSphere MQ Everyplace - Explorer (MQe_Explorer v1.2)

An WebSphere MQ Everyplace administration tool developed exclusively to support the Microsoft® Windows range of operating systems

Note: Version 1.0 of MQe_Explorer is not compatible with WebSphere MQ Everyplace Version 2.0.

The WebSphere MQ Everyplace Web pages provide full details on how to download each SupportPac.

Uninstalling WebSphere MQ Everyplace

The removal of WebSphere MQ Everyplace from your computer depends on the operating system that you are running. Follow the instructions that relate to your operating system.

On Windows

You can uninstall WebSphere MQ Everyplace from your Windows system in any of the following ways:

Using the Windows Control Panel

You can uninstall WebSphere MQ Everyplace using the standard Windows Control Panel.

1. Open the "Control Panel" window by clicking on **Start->Settings->Control Panel**.

2. Double-click on the **Add/Remove Programs** icon.
3. Scroll through the list of programs in the resulting "Add/Remove Programs" dialog box and click on "IBM WebSphereMQ Everyplace".
4. Click the **Add/Remove...** button to start the uninstall program.

Follow the on screen prompts or instructions until the program indicates that the uninstall is complete.

Using uninstall.exe

When installed on your computer, WebSphere MQ Everyplace includes a program called `uninstall.exe` that you can use to uninstall it. To uninstall WebSphere MQ Everyplace, double-click the `uninstall.exe` or use the Command:

```
<MQe directory>\Uninst\uninstall.exe
```

<MQe directory> is the directory where you installed WebSphere MQ Everyplace. This launches the Uninstall program. Follow the screen prompts or instructions until the program indicates that the uninstall is complete.

Using uninstall.jar

Use the `uninstall.jar` file as follows:

```
set classpath=<MQe directory>\Uninst\uninstall.jar;%classpath%
jview run
```

On Linux, AIX, Solaris, and HP-UX

You can uninstall WebSphere MQ Everyplace from your computer in any of the following ways:

Note: On AIX always use the uninstall launcher or jar file, *do not use SMIT*.

Using uninstall.bin

When installed on your computer, WebSphere MQ Everyplace includes a program called `uninstall.bin` that you can use to uninstall it. To uninstall WebSphere MQ Everyplace, use the Command:

```
<MQe directory>/Uninst/uninstall.bin
```

<MQe directory> is the directory where you installed WebSphere MQ Everyplace. This defaults to `/opt/MQe`, but you can change this during the installation procedure. This command launches the Uninstall program. Follow the screen prompts or instructions until the program indicates that the uninstall is complete.

Using uninstall.jar

Use the `uninstall.jar` file as follows:

```
CLASSPATH=<MQe directory>/Uninst/uninstall.jar:$CLASSPATH
export CLASSPATH
java run
```

This command launches the Uninstall program. Follow the screen prompts or instructions until the program indicates that the uninstall is complete.

Note: Do not try to uninstall using SMIT on AIX or pkgm on Solaris as this will not remove the product correctly.

Silent Uninstall

You can run the uninstaller in silent mode. This means that no panels are displayed during the uninstall and there are no prompts for input. As when running the uninstall in graphical mode, there are two ways to run a silent uninstall.

From the jar file

Run the uninstall in the same way as previously described, but append the `-silent` flag. For example:

On Windows

```
set classpath=<MQe directory>\Uninst\uninstall.jar;%classpath%
Jview run -silent
```

On Linux, AIX, Solaris, and HP-UX

```
CLASSPATH=<MQe directory>/Uninst/uninstall.jar:$CLASSPATH
export CLASSPATH
java run -silent
```

From a platform specific launcher

Add the `-is:silent` `-silent` flags. For example:

On Windows

```
uninstall.exe -is:silent -silent
```

On Linux, AIX, Solaris, and HP-UX

```
uninstall.bin -is:silent -silent
```

Using Options Files with Silent Uninstall

You can specify an options file when running the uninstall silently. The options file allows you to:

- Set the uninstall to silent
- Select which features to uninstall

The following example options file sets the uninstall to run silently, and chooses to uninstall all the features except the Documentation feature.

```
#specify silent uninstall
-silent
#set features to active
-P Java.active=true
-P Documentation.active=false
-P CBindings.active=true
-P Native.active=true
-P Palm.active=true
```

Notes:

1. Include the `.-is:silent....` flag in the options file if running the uninstall from a launcher.
2. Do not leave any blank lines in the options.txt file.

3. Start all lines with `.#...`, or a valid command.
4. You can have multiple commands on a single line.

The following examples show how to run the uninstaller with an options file:

From the jar file

```
java -cp uninstall.jar run -options C:\options.txt
```

From a launcher

```
uninstall.exe -options C:\options.txt
```

Applying maintenance to WebSphere MQ Everyplace

Maintenance updates for WebSphere MQ Everyplace are always shipped as a complete new release. There are two options when upgrading from one release to another:

Completely uninstall the current level, and install the new level in the same directory

We recommend that you keep the install package for the current level to allow it to be restored later if necessary.

Keep the existing level and install the new level into a new directory

After installation, check your classpath to ensure that the latest level of WebSphere MQ Everyplace is being invoked. If installing on Windows, make sure that you give the shortcuts folder for the new install a different name to the existing one.

For more general information on maintenance updates and their availability see the WebSphere MQ family Web page at <http://www.software.ibm.com/mqseries/>.

Chapter 2. Getting started

This chapter provides information to enable you to start using the WebSphere MQ Everyplace toolkit.

Readme file

Before proceeding, ensure you have read the Readme file, which contains last minute information about the WebSphere MQ Everyplace files that you have copied from the product CD. Carry out any extra installation procedures that might be described in the Readme.

Getting started with the Java API

Creating an initial system

The *WebSphere MQ Everyplace Introduction* manual contains a chapter headed 'Getting started with WebSphere MQ Everyplace'. The 'Using WebSphere MQ Everyplace' section of this chapter describes the different stages of use of the product, and the 'Gaining experience' section provides information on how to get an WebSphere MQ Everyplace system up and running.

Writing your own programs

When you are ready to modify your initial system and to write your own programs, use the information provided in the *WebSphere MQ Everyplace Application Programming Guide* and *WebSphere MQ Everyplace Java Programming Reference*.

Getting started with the C Bindings

To use the C Bindings, install the Java code, and C Bindings option, using the installer.

For guidance on how to get started with the C Bindings please see the first three chapters of the *WebSphere MQ Everyplace C Bindings Programming Guide*

Getting Started with WebSphere MQ Everyplace for Native Platforms

To get started with the native implementation:

1. Select the "WebSphere® MQ Everyplace for Native Platforms" option from the installer. The header files are shared between all the native platforms and the C Bindings.
2. Select the correct binary files for the platform you wish to build against.

Chapter 3. Translation

In Version 2.0 the following components of WebSphere MQ Everyplace have been translated into languages other than English:

- Graphical user interface for the mini-certificate server
- Example trace graphical user interface

Other components such as the trace messages are NLS enabled, but have not been translated.

Translated documentation

The *WebSphere MQ Everyplace Introduction* has been translated into languages other than English. These translated documents are available for download from the WebSphere MQ library Web site at www.ibm.com/software/mqseries/library/.

Trademarks

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

AIX Everyplace IBM iSeries MQSeries WebSphere z/OS zSeries

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

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

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

Index

A

about this book iii
applying maintenance 17

C

changes vii

D

documentation 12

E

environments, software 4

G

getting started 19

H

HTML documentation 13

I

installation
 AIX 7
 alternative steps 10
 C Bindings 11
 components installed 11
 documentation 12
 Java 11
 Linux 7
 migration notes 1
 native platforms 12
 Palm OS 12
 procedures 1
 silent 8
 Solaris 7
 using options files 10
 Windows 7

J

java API 19

L

licence warning v
listener architecture vii

M

message store viii
migration 1

O

operating systems, supported 4

P

persistence of listeners vii
prerequisite information 4
prerequisites 4

R

readme file 19
registry viii
remote administration viii
required operating systems 4

S

silent installation 8
software environments 4
supported operating systems 4

T

trademarks 23
translation 21

U

uninstall 14
updates vii

W

web components 14
what is new vii
writing programs 19



Part Number: CT1P5ML

GC34-6276-00



(1P) P/N: CT1P5ML

