



# WebSphere MQ Everyplace V2.0.2



---

# Contenido

## Capítulo 1. Instalación y desinstalación de Mqe . . . . . 1

Antes de la instalación . . . . .	1
Requisitos previos . . . . .	1
Licencias. . . . .	4
Instalación de Mqe . . . . .	5
Procedimiento de instalación . . . . .	5
Instalación silenciosa . . . . .	6
Opción de línea de mandatos de modalidad de accesibilidad . . . . .	8
Instalación desde un archivo zip. . . . .	8
Componentes instalados . . . . .	8
Componentes en la web . . . . .	10
Verificación de la instalación. . . . .	10
Modificación de la instalación . . . . .	11
Desinstalación de Mqe . . . . .	12
Windows . . . . .	12

Unix. . . . .	12
Desinstalación silenciosa . . . . .	13

## Capítulo 2. Aplicación de mantenimiento a Mqe . . . . . 15

Migración de 1.2.7 a 2.0 o 2.0.1 . . . . .	15
Alias en MqeFields. . . . .	15
MqeFields. . . . .	15
MqeChannel . . . . .	16
MqeAttribute . . . . .	16
Métodos y clases obsoletos . . . . .	16
Seguridad . . . . .	16
Valores de comunicación . . . . .	17

## Índice . . . . . 19



---

# Capítulo 1. Instalación y desinstalación de MQe

---

## Antes de la instalación

### Requisitos previos

#### Plataformas soportadas

Puede instalar MQe sólo en determinadas plataformas.

Para transferir los programas y las clases Java a otras plataformas, debe utilizar un programa de descarga o de transferencia de archivos adecuado (que no se proporciona).

**Directamente soportadas con soporte de instalación:** MQe se puede instalar en las plataformas siguientes con herramientas específicas de plataforma:

- Windows 2000 Server
- Windows 2000 Professional
- Windows XP Professional
- Windows 2003 Server
- AIX Versión 5.1 y Versión 5.2
- Sun Solaris Versión 8 y Versión 9
- HP-UX Versión 11i
- Linux en Intel Kernel Versión 2.2 y 2.4 (se instala mediante un archivo zip)
- Linux en zSeries Kernel Versión 2.4

iSeries es una plataforma de servidor con soporte, pero no dispone de soporte de instalación nativo. Debe utilizar, en este caso, la QShell que se proporciona como parte del sistema operativo y, a continuación, seguir las instrucciones de de instalación de UNIX mediante un archivo zip.

- iSeries Versión 5.1 y 5.2

En el caso del resto de plataformas, puede instalar MQe con un programa de transferencia de archivos adecuado (que no se proporciona).

#### Entorno Java

Los siguientes entornos Java disponen de soporte para la difusión, la ejecución y el desarrollo de MQe Versión 2.0:

- J2ME CLDC/MIDP\*
- J2ME CDC/Foundation\*
- WECE jclRM\*
- Java 1.1.8\*\*
- J2SE 1.3\*

\* o funciones superiores; \*\* se limita sólo a 4690

MQe sólo recibe soporte en JVM que lleven el logotipo de Sun "Java Compatible" o "Java Powered".

Los siguientes sistemas operativos son las plataformas de referencia para Java. Si se experimenta un problema en el entorno Java, puede que IBM solicite reproducir el problema en una de las plataformas siguientes:

- J2ME, CLDC/MIDP, J2M3 CDC/Foundation, WECE
  - PocketPC 2002 (ARM)
  - PocketPC 2003 (ARM)
- J2ME, CLDC/MIDP, J2M3 CDC/Foundation, WECE, J2SE 1.4.2
  - Windows 2000 Professional
  - Windows XP Professional
  - Linux Intel Kernel 2.4
  - Linux Intel Kernel 2.2
- Java 1.1.8 (4690)
- J2SE 1.4.2
  - Windows 2000 Server
  - Windows 2003 Server
  - AIX 5.1
  - AIX 5.2
  - 4690
  - Solaris 8
  - Solaris 9
  - HPUX11i
  - Linux
  - zSeries V2R4
  - iSeries 5.2
  - iSeries 5.3

Para enviar mensajes a WebSphere MQ y recibirlos, son obligatorias las clases Java de MQ.

- Se incluyen en un paquete dentro de MQ Versión 5.3 y posteriores.
- Si va a utilizar una versión anterior de MQ, puede descargar las clases Java de MQe como SupportPac ma88 del sitio web de IBM MQ en  
<http://www.ibm.com/software/integration/support/supportpacs/product.html#wmq>

Para obtener más información sobre los requisitos del sistema y la implementación de Java, consulte <http://www.ibm.com/software/integration/wmqe/requirements/>

**J2ME:** MQe es compatible con las tecnologías J2ME:

1. CDC (Connected Device Configuration)
2. CLDC (Connected Limited Device Configuration)

O, más concretamente, sus perfiles más conocidos:

1. CDC/Foundation
2. CLDC/Midp

Cuando tenga que decidir cuál debe utilizar, tenga en cuenta lo siguiente:

1. Si se utiliza CDC/Foundation, podrá utilizar todas las funciones de MQe, excluidos los códigos de ejemplo que exijan el paquete de GUI de AWT.
2. Sin embargo, el uso de CLDC/Midp restringe la aplicación únicamente al comportamiento del lado del 'cliente' y a un rango limitado de cifradores y compresores integrados.

**Nota:** Los dispositivos con soporte de CLDC/Midp pueden tener graves limitaciones de memoria que pueden descartar el uso directo del paquete principal de MQe.

**Entorno de enlaces C:** Las API de los enlaces C exigen J2SE 1.3 o posterior y sólo se pueden instalar en:

- Windows 2000 Server
- Windows 2003 Server
- Windows 2000 Professional
- Windows XP Professional

**Interfaz JMX:** La interfaz de MQe Java Management Extensions (MQe JMX) se ejecuta como una aplicación en una Java Virtual Machine (JVM). Lo único que exige es un gestor de colas local que esté activado. Por ello, la interfaz puede gestionar el gestor de colas local instrumentado y los recursos del gestor de colas. También puede gestionar de forma remota todos los gestores de colas de MQe activados (y sus recursos) para los que el gestor de colas local pueda conectarse directamente a la red de MQe.

La API de MQe JMX se ha desarrollado conforme a la especificación JMX v.1.2 y, por lo tanto, exige una implementación compatible de la especificación JMX. Si no dispone de esa implementación, Sun proporciona una implementación de referencia de JMX en:  
<http://java.sun.com/products/JavaManagement/>.

## Entorno nativo C

Para obtener información general, consulte el manual *Consulta de programación en C*, concretamente la página *Información de compilación*, aunque esa página ha quedado un poco desfasada y en este tema se proporciona una actualización.

En el caso de la base de código en C, se proporciona soporte para cinco plataformas:

- PocketPC2000
- PocketPC2002
- PocketPC2003
- WinCE .NET 4.2
- Windows de 32 bits.

Se proporciona soporte para las DLL en C de WebSphere MQ Everyplace en WinCE .NET 4.2 mediante los servicios de invocación de plataforma que proporciona el entorno .NET.

En el caso de PocketPC, se proporcionan archivos binarios tanto para el dispositivo como para el emulador disponible en el entorno de desarrollo integrado Microsoft Embedded Visual C++. Los archivos binarios que se proporcionan para los dispositivos se compilan para procesadores ARM.

### Archivos binarios

La raíz de los archivos binarios, así como de la documentación y los ejemplos, es el directorio C que depende del directorio que haya elegido para instalar MQe.

En el directorio C, los archivos se encuentran en las siguientes ubicaciones:

#### PocketPC2000

##### ARM

**DLL** C:\PocketPc2000\arm\bin

**LIB** C:\PocketPc2000\arm\lib

##### Emulador

**DLL** C:\PocketPc2000\x86emulator\bin

**LIB** C:\PocketPc2000\x86emulator\lib

#### PocketPC2002

##### ARM

DLL C:\PocketPc2002\arm\bin

LIB C:\PocketPc2002\arm\lib

#### Emulador

DLL C:\PocketPc2002\x86emulator\bin

LIB C:\PocketPc2002\x86emulator\lib

### PocketPC2003 y WinCE .NET 4.2

#### ARM

DLL C:\PocketPc2003\arm\bin

LIB C:\PocketPc2003\arm\lib

#### Emulador

DLL C:\PocketPc2003\x86emulator\bin

LIB C:\PocketPc2003\x86emulator\lib

### Windows de 32 bits

DLL C:\Win32\Native\bin

LIB C:\Win32\Native\lib

### Archivos de cabecera

Los archivos de cabecera son comunes a todas las plataformas nativas y puede encontrarlos en el directorio include que depende del directorio de instalación.

#### MQe\_API.h

Éste es el archivo de cabecera "raíz". Si se incluye, todos los archivos de cabecera relevantes se incluirán de forma automática.

Para garantizar que se incluyan los archivos y las definiciones correctas, debe indicar que está ejecutando la base de código nativa de la manera siguiente:

```
#define NATIVE // o especifíquelo como una opción para el compilador
#include <published/MQe_API.h>
```

### Enlaces

Debe enlazar con las dos bibliotecas siguientes:

#### HMQ\_nativeAPI.lib

// biblioteca de API

#### HMQ\_nativeCnst.lib

// biblioteca de MQeString constante estática

Por lo general, debe incluir estos dos archivos. A continuación, un enlazador de optimización debe eliminar los enlaces con las funciones y constantes que no haya utilizado.

Las otras bibliotecas de MQe enlazan de forma estática y dinámica con la biblioteca de API principal y se incluirán según sea necesario.

## Licencias

MQe es un kit de herramientas que permite a los usuarios escribir aplicaciones de MQe y crear un entorno en el que ejecutarlas. Antes de difundir este producto, o las aplicaciones que lo utilizan, asegúrese de tener las licencias necesarias.

1. El precio de las licencias para la utilización del programa en **servidores** se basa en *Unidades de licencia de procesador*. El uso de una copia del programa en un servidor exige la adquisición de una *Unidad de licencia de procesador* para cada procesador o multiprocesador simétrico que contenga el

servidor en el que se ejecute la copia del programa. Se necesitan distintos tipos de *Unidades de licencia de procesador* y de *Autorizaciones de uso de dispositivo* en función de si el programa se ejecuta en punto de venta, es decir, al por menor, en equipo o en otro tipo de sistema. Para utilizar el programa en un equipo minorista es necesaria una licencia de servidor *Minorista*, mientras que para utilizarlo en otro equipo (no-minorista) es necesaria una licencia de servidor de *Red*.

2. Se exige una *Autorización de uso de dispositivo* adicional para la utilización del Programa en un **dispositivo cliente** independiente, excepto los incluidos en la *Licencia de servidor de Red* que se describe en el punto 3 a continuación.
3. Cada *Licencia de servidor de red* incluye autorización para el uso restringido del Programa con un número no superior a cien (100) dispositivos cliente, con la condición de que todas estas copias se utilicen en la misma empresa económica u organización que la copia del servidor.

Consulte <http://www.ibm.com/software/integration/wmqe/> para obtener detalles sobre estas restricciones.

Se exigen *Autorizaciones de uso de plataforma de dispositivo*, que se registran en los documentos del Comprobante de titularidad y son válidas para dar soporte al uso de MQE, para poder utilizar el producto (con una finalidad que no sea el desarrollo y prueba de código) en las plataformas cliente especificadas. Estas licencias no permiten al usuario utilizar el puente de MQE ni trabajar en las plataformas de servidor que se especifican en las listas de grupos de precios de MQE que publica IBM y que también están disponibles en Internet en el URL que se indica a continuación.

Consulte <http://www.ibm.com/software/integration/mqfamily/> para obtener detalles sobre estas restricciones.

---

## Instalación de MQE

La información de este capítulo le guiará durante la instalación de MQE en máquinas que vayan a utilizarse para desarrollar aplicaciones de MQE.

El programa de instalación de MQE es un archivo jar Java que tiene programas de ejecución específicos de cada plataforma, los cuales pueden ejecutarse directamente desde el CD del producto. El programa de instalación extrae los archivos de trabajo en un directorio temporal, copia los archivos de MQE en la máquina y borra los archivos de trabajo.

Tenga en cuenta que, en este release, el proveedor de la aplicación y de la solución es el responsable de difundir MQE en dispositivos de tecnología de dispositivos ubicuos.

## Procedimiento de instalación

La información de este apartado es aplicable a la instalación en Windows, AIX, Linux, Solaris y HP-UX.

En cualquier momento durante la instalación puede pulsar el botón **Anterior** en cualquier pantalla para regresar a la pantalla anterior, donde puede revisar o modificar la información. Para salir del procedimiento de instalación y cancelarlo, pulse el botón **Cancelar** en cualquier pantalla.

### Antes de empezar:

- Es muy aconsejable desinstalar las versiones anteriores de MQE antes de instalar o volver a instalar esta nueva versión (consulte “Desinstalación de MQE” en la página 12). El programa de instalación no detecta versiones del producto anteriores a la Versión 1.2.4 y no muestra avisos.
- Si está utilizando **Windows**, compruebe si su ID de usuario tiene acceso como administrador. Si no es así, es posible que los iconos del menú Inicio de IBM MQE no se visualicen.
- Si utiliza **AIX**, ha de iniciar la sesión como usuario root para que la instalación se ejecute correctamente.

## Para instalar MQe:

1. Inserte el CD del producto en su unidad de CD-ROM.
2. Inicie la instalación desde el programa de ejecución específico de la plataforma o desde el archivo setup.jar.

- **Instalación desde el programa de ejecución específico de la plataforma:**

- a. Los programas de ejecución se encuentran en los subdirectorios específicos de cada plataforma, en el CD del producto. Para iniciar el proceso de instalación, ejecute el programa de ejecución correcto para la plataforma (por ejemplo, setup.exe para Windows).
- b. Si copia el programa de ejecución en la máquina local de destino, también deberá copiar el archivo setup.jar en el directorio padre del directorio en el que haya copiado el programa de ejecución. El programa de ejecución no se puede ejecutar desde un directorio raíz.
- c. Si ejecuta la instalación desde un programa de ejecución y ve un recuadro de mensaje con el texto No se ha encontrado ninguna JVM coincidente, significa que el programa de instalación no ha encontrado un entorno Java que pueda utilizar. Si ve este mensaje, necesitará utilizar el archivo setup.jar para la instalación.
- d. Para indicar al programa de ejecución que utilice una JVM específica, utilice el siguiente distintivo:

```
setup.exe -is:javahome c:\jdk1.3
```

Para indicar al programa de ejecución que utilice un directorio específico para almacenar los archivos temporales, utilice el siguiente distintivo:

```
setup.exe -is:tempdir c:\midirtemp
```

- **Desde el archivo setup.jar:**

Vaya al directorio del CD del producto en el que se encuentre setup.jar y ejecute el programa de instalación utilizando el mandato Java en su máquina. Este mandato suele ser java, jre o jview. Por ejemplo:

### En Windows

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

(Si utiliza JVM 1.2.2 o superior, puede ejecutar el archivo jar efectuando una doble pulsación sobre el mismo.)

### En Linux, AIX, Solaris y HP-UX

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

3. En la pantalla de bienvenida, confirme que desea instalar el programa pulsando el botón **Siguiente** y siguiendo las indicaciones para llevar a cabo la instalación.

## Instalación silenciosa

Puede ejecutar el programa de instalación en modalidad silenciosa. Esto significa que no aparecen paneles durante la instalación y no se solicita información al usuario. Existen dos maneras de ejecutar la instalación de forma silenciosa:

### Desde el archivo jar

Ejecute la instalación del modo que se ha descrito anteriormente, pero añada el distintivo -silent. Por ejemplo:

#### En Windows

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

#### En Linux, AIX, Solaris y HP-UX

```

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

```

### Desde un programa de ejecución específico de la plataforma

Añada los distintivos `-is:silent -silent`. Por ejemplo:

#### En Windows

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

#### En AIX

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

#### En Linux

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

#### En Solaris

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

#### En HP-UX

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

## Directorios de instalación silenciosa

Por omisión, el programa de instalación instala MQe en los directorios que se muestran en la tabla siguiente. Si tiene versiones anteriores de MQe en el sistema, en las instalaciones silenciosas se utiliza el directorio actual y no el directorio por omisión que se muestra en esta tabla:

Tabla 1. Directorios de instalación por omisión

Plataforma	Directorio de instalación por omisión
iSeries	/opt/MQe
AIX	/opt/MQe
Linux	/opt/MQe
Solaris	/opt/MQe
Win32	\Archivos de programa\MQe
HP-UX	/opt/MQe

Para definir un directorio de instalación diferente en una instalación silenciosa, añade el distintivo `-P MQe.installLocation` al mandato de instalación del siguiente modo:

#### Desde el archivo jar

```
java run -silent -P MQe.installLocation="C:\nuevo directorio de instalación"
```

#### Desde un programa de ejecución específico de la plataforma

```

setup.exe -is:silent -silent -P MQe.installLocation=
"C:\nuevo directorio de instalación"

```

**Nota:** Si utiliza el parámetro `-P MQe.installLocation` para modificar la ubicación de instalación durante una instalación silenciosa, asegúrese de especificar la vía de acceso completa del nuevo destino.

## Instalación silenciosa con archivos de opción

Cuando ejecute una instalación silenciosa, puede especificar un archivo de opciones. El archivo de opciones le permite:

- Definir la instalación como silenciosa
- Cambiar la ubicación de la instalación
- Seleccionar las funciones que desea instalar

En el archivo de opciones de ejemplo siguiente se establece que la instalación se ejecute de forma silenciosa, que la ubicación de instalación sea C:\MQe y se selecciona la instalación de todos los componentes.

```
#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"
```

#### Nota:

1. Incluya el indicador `-is:silent` en el archivo de opciones si está efectuando la instalación desde el iniciador.
2. No deje ninguna línea en blanco en el archivo `options.txt`.
3. Todas las líneas han de comenzar por `#` o un mandato válido.
4. Puede tener varios mandatos en una sola línea.

Los ejemplos siguientes muestran cómo ejecutar el programa de instalación con un archivo de opciones:

#### Desde el archivo jar

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

#### Desde un programa de ejecución

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

## Opción de línea de mandatos de modalidad de accesibilidad

La opción de línea de mandatos de modalidad de accesibilidad establece que la instalación se lleve a cabo de forma automática en modalidad de consola y proporciona información adicional en audio para usuarios finales con discapacidad visual. Ejemplo de uso:

```
java -cp <classpath> setup.jar run -accessibility
```

Este distintivo también es válido durante la desinstalación. Ejemplo de uso:

```
java -cp <classpath> uninstall.jar run -accessibility
```

## Instalación desde un archivo zip

Las clases de MQe también se proporcionan como archivo zip. Puede utilizar este archivo para instalar MQe en dispositivos en los que no se dé soporte al programa de instalación gráfico o bien no resulte adecuado. En un sistema basado en UNIX (como Linux y HP-UX), tiene que crear una carpeta, copiar en ella el archivo zip adecuado y luego utilizar un programa de utilidad de descompresión para extraer los archivos de clase. Por ejemplo:

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

Tras extraer los archivos de clase, configure el entorno para que ejecute los programas de MQe.

## Componentes instalados

Puede seleccionar varias funciones durante la instalación de MQe. Estas funciones se describen a continuación, junto con los componentes que se instalan si se selecciona la función.

## MQe para Java

### Clases MQe Java

Conjunto de clases que implementan todas las funciones de MQe. Se pueden utilizar subconjuntos de estas clases para proporcionar configuraciones diferentes de MQe, como un subconjunto para un dispositivo o un subconjunto para un servidor.

### Clases de ayuda

Un conjunto de clases derivado de las clases básicas que implementan algunas de las funciones que se utilizan normalmente.

### Clases de ejemplo

Conjunto de clases que muestran cómo utilizar muchas de las funciones de MQe. También se proporciona el código fuente de estos archivos de clase.

### Programas de utilidad

Herramientas que ayudan a programar y administrar MQe.

## Enlaces C de MQe

### Clases específicas de los enlaces C

Incluye el paquete `com.ibm.mqe.bindings`. Sólo se necesitan para los enlaces C y no afectan a las funciones existentes.

### Archivos de cabecera y binarios

Enlaces C añade archivos binarios adicionales y archivos de cabecera. Encontrará información completa sobre estos archivos en la publicación *MQe C Bindings Programming Guide*.

### MQe C Bindings Programming Guide SC34-6280-01

Este manual contiene una guía e información sobre procedimientos para escribir aplicaciones de MQe en C y para administrar sistemas. El nombre del archivo es `hmq9a1_WMQE_C_BindingsProgrammingGuide.pdf`.

Consulte también la versión en html de la guía de programación de enlaces C.

## MQe para plataformas nativas

### Archivos de cabecera

El conjunto de archivos de cabecera necesarios es común para todas las plataformas nativas y también se comparte con los enlaces C.

### Archivos binarios

Hay archivos DLL y LIB, creados contra los siguientes SDK:

- PocketPC 2002 SDK, procesador ARM y emulador PocketPC 2002
- PocketPC 2003 SDK, procesador ARM y emulador PocketPC 2003

### Ejemplos

Algunos ejemplos también están documentados en el manual *MQe Native Platforms Programming Reference*.

## Herramienta de información de versión PocketPC

La herramienta de información `MQeVersion` se envía como parte de la base del código nativo. Se trata de un ejecutable que permite comprobar la información de la versión correspondiente a los archivos DLL nativos instalados. Esta herramienta se ha creado para PocketPC 2000, 2002 y 2003.

El archivo del programa se encuentra en el directorio `C:\tools\Version\PocketPc2000\arm o C:\tools\Version\PocketPc2002\arm` de la instalación. Es necesario copiar este archivo en un lugar adecuado del dispositivo. Al ejecutarse, la herramienta intenta cargar todos los archivos DLL nativos. En el escribir del dispositivo se graba un archivo con información sobre los archivos DLL.

- Ejecute la herramienta directamente desde el explorador de archivos o una línea de mandatos, sin argumentos y la herramienta cargará los archivos DLL que se encuentren utilizando la vía de acceso de carga estándar del dispositivo.
- Ejecute la herramienta en una línea de mandatos para especificar un directorio como parámetro. La especificación depende parcialmente del dispositivo, pero puede optar por utilizar uno de los intérpretes de línea de mandatos disponibles para el dispositivo PocketPC. Cuando se especifica un directorio, sólo se comprueban los archivos DLL que se encuentran en dicho directorio.

Para cada archivo DLL, la herramienta facilita el número de versión principal, un posible nivel de arreglo electrónico, es decir, el cuarto dígito del número de versión, y el ID del build del que procede el archivo DLL. Si hay archivos DLL que ha decidido no copiar en el dispositivo puede ver informes de mensajes relacionados con esto.

**Nota:** Si ve archivos DLL procedentes de otros builds o versiones, compruebe con cuidado si los archivos que ha copiado son correctos. Dentro de la instalación hay archivos DLL para los emuladores de PocketPC así como de los dispositivos. Asegúrese de que copia los archivos correctos. Si no lo hace, se producirán errores cuando intente ejecutar los programas.

## Documentación

Los manuales que se entregan con las versiones anteriores de MQe se han modificado y reestructurado sustancialmente en este centro de información de MQe.

## Componentes en la web

Normalmente se desea instalar más de un producto MQe a fin de desarrollar aplicaciones.

## Verificación de la instalación

En este apartado se describe cómo ejecutar algunos ejemplos en los que se verifica la instalación correcta de un kit de desarrollo de MQe.

### Verificación de la instalación de Java

Después de instalar MQe, puede utilizar los procedimientos siguientes para ejecutar algunos ejemplos que determinan si la instalación del kit de desarrollo ha sido satisfactoria.

- Asegúrese de que el entorno Java se haya configurado como se describe en “Entorno Java” en la página 1. Al ejecutar cualquiera de los archivos de proceso por lotes de Windows descritos en este apartado, el primer parámetro de cada uno es el nombre del Java Development Kit que se debe utilizar. Si no se especifica un nombre, el valor por omisión es IBM.

**Nota:** los scripts de shell de UNIX no tienen un parámetro correspondiente.

- Vaya al directorio correcto:

#### Windows

Vaya al directorio <directorio\_instalación\_MQe>\Java\demo\Windows.

**UNIX** Vaya al directorio <directorio\_instalación\_MQe>/Java/demo/UNIX.

- Cree un gestor de colas del modo siguiente:

#### Windows

Ejecute el archivo por lotes:

Ejemplo en C

CreateExampleQM.bat <JDK>

**UNIX** Ejecute el script de shell:

Ejemplo en C

CreateExampleQM

para crear un gestor de colas de ejemplo llamado ExampleQM.

Una parte del proceso de creación es establecer los directorios que contendrán la información de configuración del gestor de colas y las colas. El ejemplo utiliza un directorio relativo al directorio actual denominado ExampleQM. En este directorio existen dos directorios más:

- Registro - contiene archivos que tienen datos de configuración de gestor de colas.
  - Colas - para cada cola, hay un subdirectorío para contener los mensajes de la cola. (El directorio no se crea hasta que se activa la cola).
- Ejecute una aplicación sencilla del modo siguiente:

Una vez se haya creado un gestor de colas, éste podrá iniciarse y se podrá utilizar en aplicaciones. Puede utilizar el archivo de proceso por lotes ExamplesMQeClientTest.bat o el script de shell ExamplesMQeClientTest para ejecutar algunos de los ejemplos de aplicación sencillos.

El archivo de proceso por lotes ejecuta examples.application.Example1 por omisión. Este ejemplo transfiere un mensaje de prueba a un gestor de colas ExampleQM y, a continuación, obtiene el mensaje del mismo gestor de colas. Si los dos mensajes coinciden, la aplicación se ha ejecutado correctamente.

En el paquete examples.application se incluye un conjunto de aplicaciones que muestran las diferentes funciones de MQe. Se pueden ejecutar dichos ejemplos del modo siguiente:

#### Windows

Pasar parámetros a los archivos por lotes:

Ejemplo en C

```
ExamplesMQeClientTest <JDK> <ExampleNo>
```

UNIX Pasar parámetros a los scripts de shell:

Ejemplo en C

```
ExamplesMQeClientTest &gt;ExampleNo&lt;
```

donde *Ejemplo\_núm.* es el sufijo del ejemplo. Puede ir de 1 a 6.

- Suprima un gestor de colas.

Cuando ya no sea necesario un gestor de colas, éste se puede suprimir. Para suprimir el gestor de colas de ejemplo ExampleQM:

#### Windows

Ejecute el archivo por lotes

Ejemplo en C

```
DeleteExampleQM.bat <JDK>
```

UNIX Ejecute el script de shell

Ejemplo en C

```
DeleteExampleQM
```

Una vez se haya suprimido el gestor de colas, éste no podrá iniciarse.

**Nota:** Los ejemplos utilizan directorios relativos para facilitar la configuración. Se recomienda utilizar directorios absolutos para otras tareas que no sean las tareas de desarrollo base y demostración.

Cuando esto sucede y se están utilizando directorios relativos, el gestor de colas ya no puede localizar su información de configuración ni sus colas.

## Verificación de la instalación de C

Después de instalar MQe, puede ejecutar algunos ejemplos de la consulta de programación en C de MQe para verificar la instalación.

---

## Modificación de la instalación

Si desea eliminar la instalación, consulte “Desinstalación de MQe” en la página 12. Tenga en cuenta que no es posible utilizar *Agregar o quitar programas* para modificar las opciones que haya instalado, sino sólo para desinstalar el producto completamente.

Para modificar la instalación (por ejemplo, para añadir una función o un componente que no haya seleccionado durante la instalación), inserte el CD del producto en la unidad y, a continuación, siga las instrucciones de "Procedimiento de instalación" en la página 5.

---

## Desinstalación de MQe

Siga las instrucciones que correspondan al sistema operativo que utilice.

### Windows

Elija una de las maneras siguientes para desinstalar MQe del sistema Windows:

#### Utilización del Panel de control de Windows

1. Pulse **Inicio->Configuración->Panel de control**.
2. Pulse dos veces el icono **Agregar o quitar programas**.
3. En el diálogo *Agregar o quitar programas*, pulse **IBM MQe**.
4. Pulse el botón **Agregar o quitar...** para iniciar el programa de desinstalación.

Siga las indicaciones o instrucciones de la pantalla hasta que el programa indique que la desinstalación ha finalizado.

#### Utilización de uninstall.exe

En el <directorio de instalación de MQe>, pulse dos veces `uninstall.exe` o utilice el mandato:  
<directorio de instalación de MQe>\Uninst\uninstall.exe

donde <directorio instalación de MQe> es el directorio donde ha instalado MQe.

Siga las indicaciones hasta que el programa indique que la desinstalación ha finalizado.

#### Utilización de uninstall.jar

Utilice el archivo `uninstall.jar` del modo siguiente:

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

### Unix

Siga estas instrucciones para desinstalar MQe en:

- AIX
- HP-UX
- Linux
- Solaris
- Unix

Elija una de las maneras siguientes para desinstalar MQe del sistema:

**Nota:** en AIX, utilice siempre uno de estos métodos; *no utilice SMIT* porque no eliminará el producto correctamente.

**Nota:** en Solaris, utilice siempre uno de estos métodos, *no utilice pkgrm* porque no eliminará el producto correctamente.

## Mediante uninstall.bin

Entre el mandato:

```
<directorio de instalación de MQe>/Uninst/uninstall.bin
```

<directorio de instalación de MQe> es el directorio en el que ha instalado MQe. El valor por omisión es /opt/MQe, pero puede cambiarlo durante el procedimiento de instalación.

Siga las indicaciones hasta que el programa indique que la desinstalación ha finalizado.

## Utilización de uninstall.jar

Utilice los mandatos siguientes para invocar uninstall.jar:

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

Siga las indicaciones hasta que el programa indique que la desinstalación ha finalizado.

## Desinstalación silenciosa

Puede ejecutar el programa de desinstalación en modalidad silenciosa. Esto significa que no aparecen paneles durante la desinstalación y no se solicita información al usuario. Existen dos maneras de ejecutar la desinstalación de forma silenciosa:

### Desde el archivo jar

Ejecute la desinstalación del modo que se ha descrito anteriormente, pero añada el distintivo `-silent`. Por ejemplo:

#### En Windows

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

#### En Linux, AIX, Solaris y HP-UX

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

### Desde un programa de ejecución específico de la plataforma

Añada los distintivos `-is:silent -silent`. Por ejemplo:

#### En Windows

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

#### En Linux, AIX, Solaris y HP-UX

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

## Desinstalación silenciosa con archivos de opciones

Cuando ejecute una desinstalación silenciosa, puede especificar un archivo de opciones. El archivo de opciones le permite:

- Definir la desinstalación como silenciosa
- Seleccionar las funciones que desea desinstalar

El siguiente archivo de opciones de ejemplo define que la desinstalación se ejecute de forma silenciosa y elige desinstalar todas las funciones excepto la Documentación.

```
#especifica una desinstalación silenciosa
-silent
#define las funciones activas
-P Java.active=true
-P Documentation.active=false-P CBindings.active=true
-P Native.active=true
```

**Nota:**

1. Incluya el indicador `-is:silent` en el archivo de opciones si está efectuando la desinstalación desde un iniciador.
2. No deje ninguna línea en blanco en el archivo `options.txt`.
3. Todas las líneas deben empezar por `#...` o un mandato válido.
4. Puede tener varios mandatos en una sola línea.

Los siguientes ejemplos muestran cómo ejecutar el programa de desinstalación con un archivo de opciones:

**Desde el archivo jar**

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

**Desde un programa de ejecución**

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

---

## Capítulo 2. Aplicación de mantenimiento a MQe

Las actualizaciones de mantenimiento de MQe se entregan como un nuevo release completo.

Existen dos opciones para realizar una actualización de un release a otro:

### Desinstalar por completo el nivel actual e instalar el nuevo nivel en el mismo directorio

Mantener el paquete de instalación del nivel actual, en caso de que desee restaurarlo más adelante

### Conservar el nivel existente e instalar el nuevo nivel en un nuevo directorio

Tras la instalación, compruebe la variable `classpath` para asegurarse de que se invoque el nivel más reciente de MQe. Si realiza la instalación en Windows, asegúrese de asignar otro nombre a la carpeta de métodos abreviados de la nueva instalación.

Para obtener información más general sobre las actualizaciones de mantenimiento y su disponibilidad, consulte la página web de la familia MQ en <http://www.ibm.com/software/integration/mqfamily/>.

---

## Migración de 1.2.7 a 2.0 o 2.0.1

Cuando realice una actualización a la Versión 2.0 o a la Versión 2.0.1, debe tener en cuenta el modo en que los cambios descritos en este apartado afectarán a la aplicación de MQe.

### Alias en MQeFields

En la Versión 1, la estructura MQeFields que se pasaba a MQeQueueManager permitía especificar los dos alias siguientes:

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

Estos alias especificaban los nombres de clases por omisión que debían utilizarse al cargar claves de atributo, si no se había especificado una clase de clave de atributo.

**Estos valores están codificados de forma permanente en la base del código de la Versión 2.0 y 2.0.1 y no se pueden modificar con el mecanismo de alias. Si los valores se especifican en archivos .ini o en invocaciones a MQeQueueManager, se ignorarán.**

### MQeFields

Para cumplir con la especificación Java 2 Platform Micro Edition's (J2ME) Connected Limited Device Configuration (CLDC)/Mobile Information Device Protocol (MIDP), se han modificado o eliminado varios métodos de MQeFields:

- El uso explícito de los tipos de punto flotante, `float` y `double` se ha eliminado.

Por ejemplo, puede que haya utilizado `putFloat("Val1", -1.234)`.

En plataformas Java que permiten el uso de `float/double`, esta funcionalidad se puede emular convirtiendo explícitamente los datos en el equivalente `int` o `long` utilizando los tipos base del método `Java Object Convert`.

En ese caso, el método anterior se sustituye por `putFloatAsInt("Val1", Float.floatToIntBits(-1.234))`.

**Nota:** Las aplicaciones de la versión 1 pueden recuperar estos valores de forma normal.

- Los métodos `dumpToFile` y `restoreFromFile` se han eliminado.

Las aplicaciones que utilizaban esas funciones ahora tienen que hacer un vuelco del objeto MQeFields y escribir la matriz de bytes en el archivo especificado.

- La ejecución Xor de datos volcados también se ha eliminado.

## MQeChannel

La clase

`com.ibm.mqe.MQeChannel`

se ha cambiado de ubicación y ahora se denomina

`com.ibm.mqe.communications.MQeChannel`

Cualquier referencia que se haga al nombre anterior de esta clase en los mensajes de administración se sustituirá automáticamente por el nuevo nombre de la clase.

## MQeAttribute

Los siguientes cambios se han implementado en relación con MQeAttribute:

- El nombre de la implementación del método `equals()`

en MQeAttribute y sus subclases en la Versión 1.2.7 (y versiones anteriores) se ha cambiado por `isAcceptable()`

- Actualmente, se envía una norma MQeAttributeRule con el producto.

Ahora debería ampliar sus normas de atributos de esta clase en lugar de MQeRule.

Todos los métodos de MQeAttribute y sus subclases, que solían tomar un objeto MQeRule como uno de sus parámetros, ahora toman en su lugar un objeto MQeAttributeRule.

## Métodos y clases obsoletos

Las clases que aparecen aquí se han eliminado del producto.

Recomendamos actualizar las aplicaciones que se hayan escrito para utilizar estas clases en lugar de utilizar la función equivalente que se proporciona en MQe Versión 2.0 y Versión 2.0.1.

Para permitir que las aplicaciones existentes se ejecuten antes de actualizarlas, MQe proporciona el archivo jar MQeDeprecated.jar.

El archivo MQeDeprecated.jar contiene las siguientes clases:

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

Si desea ver información más detallada sobre las sustituciones de las clases indicadas, consulte la lista para cada clase en la consulta de programación en Java.

## Seguridad

Se han realizado los siguientes cambios en la seguridad:

1. Las clases MQeCL y MQeRandom se han sustituido por la clase CL de cryptoLite.
2. El antiguo soporte de minicertificados se ha retirado en la Versión 2.0 y posteriores.

Estos cambios tienen las siguientes implicaciones:

1. La clase CL se suministra como cryptoLite.zip. Para poder utilizar la seguridad de MQe, el archivo zip debe estar en la variable classpath de Java.
2. MQeMiniCertificateServer ya no da soporte al antiguo minicertificado.

## **Valores de comunicación**

En la versión 2, el tiempo de espera del canal por omisión se ha reducido de 1 hora a 5 minutos. Además, las propiedades del sistema que se utilizan para especificar ciertas variables de adaptador (por ejemplo, tiempos de espera de socket) se han modificado.

Esto puede afectar a los clientes que trabajen en redes lentas o frágiles.



---

# Índice

## A

actualización 15  
aplicación de mantenimiento 15

## E

entornos, software 1  
entornos de software 1

## I

información necesaria 5

instalación  
notas sobre migración 5

## M

mantenimiento, aplicar 15  
migración 5

## R

requisitos previos 1

## S

sistemas operativos, soportados 1  
sistemas operativos requeridos 1  
sistemas operativos soportados 1

## V

verificar la instalación  
C 10  
Java 10