



# WebSphere MQ Everyplace V2.0.2



---

# Inhaltsverzeichnis

## Kapitel 1. MQe installieren und deinstallieren . . . . . 1

Vor der Installation . . . . .	1
Voraussetzungen . . . . .	1
Lizenzierung . . . . .	5
Installation von MQe . . . . .	5
Installationsvorgang . . . . .	5
Unbeaufsichtigte Installation . . . . .	7
Befehlszeilenoption für den Eingabehilfenmodus . . . . .	8
Von einer ZIP-Datei aus installieren. . . . .	9
Installierte Komponenten . . . . .	9
Komponenten im Web. . . . .	10
Installation überprüfen . . . . .	10
Installation modifizieren . . . . .	12
Deinstallation von MQe . . . . .	12
Windows . . . . .	12

Unix. . . . .	13
Unbeaufsichtigte Deinstallation. . . . .	14

## Kapitel 2. Wartung von MQe . . . . . 15

Migration von 1.2.7 auf 2.0 oder 2.0.1. . . . .	15
Aliasnamen in MQeFields . . . . .	15
MQeFields. . . . .	15
MQeChannel . . . . .	16
MQeAttribute . . . . .	16
Nicht weiter unterstützte Methoden und Klassen . . . . .	16
Sicherheit . . . . .	17
Übertragungseinstellungen . . . . .	17

## Index . . . . . 19



---

# Kapitel 1. MQe installieren und deinstallieren

---

## Vor der Installation

### Voraussetzungen

#### Unterstützte Plattformen

MQe kann nur auf bestimmten Serverplattformen installiert werden.

Wenn Programme und Java-Klassen auf andere Plattformen übertragen werden sollen, muss ein entsprechendes Programm zum Herunterladen oder Übertragen von Dateien (nicht im Lieferumfang enthalten) verwendet werden.

**Direkt unterstützte Plattformen mit Installationsunterstützung:** MQe kann auf den folgenden Plattformen mit plattformspezifischen Tools installiert werden:

- Windows 2000 Server
- Windows 2000 Professional
- Windows XP Professional
- Windows 2003 Server
- AIX Version 5.1 und Version 5.2
- Sun Solaris Version 8 und Version 9
- HP-UX Version 11i
- Linux on Intel Kernel Version 2.2 und 2.4 (unter Verwendung einer ZIP-Datei installiert)
- Linux on zSeries Kernel Version 2.4

iSeries ist eine unterstützte Serverplattform, hat aber keine native Installationsunterstützung. Stattdessen müssen Sie die als Teil des Betriebssystems zur Verfügung stehende QShell verwenden und dann der Anleitung für UNIX zur Installation unter Verwendung einer ZIP-Datei folgen.

- iSeries Version 5.1 und 5.2

Bei allen übrigen Plattformen können Sie MQe mit Hilfe eines geeigneten Dateiübertragungsprogramms (nicht im Lieferumfang enthalten) installieren.

#### Java-Umgebung

Die folgenden Java-Umgebungen werden bei der Implementierungskomponente, der Laufzeitkomponente und der Entwicklungskomponente von MQe Version 2.0 unterstützt:

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

\* auch größerer Funktionsumfang; \*\* beschränkt auf 4690

MQe wird nur auf JVMs mit dem Sun-Logo "Java Compatible" oder "Java Powered" unterstützt.

Die folgenden Betriebssysteme sind die Referenzplattformen für Java. Wenn ein Problem innerhalb einer Java-Umgebung auftritt, bittet IBM unter Umständen um dessen Reproduktion auf einer der folgenden Plattformen:

- 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

Zum Senden von Nachrichten an WebSphere MQ und zum Empfangen von Nachrichten von WebSphere MQ sind die MQ-Klassen für Java erforderlich.

- Diese sind bei MQ Version 5.3 und späteren Versionen mit im Paket enthalten.
- Wenn Sie eine frühere Version von MQ verwenden, können Sie die MQe-Klassen für Java als Support-Pac ma88 von der MQ-Website von IBM unter folgender Adresse herunterladen:

<http://www.ibm.com/software/integration/support/supportpacs/product.html#wmq>

Weitere Informationen zu Systemanforderungen und Java-Implementierung finden Sie im Abschnitt <http://www.ibm.com/software/integration/wmqe/requirements/>.

**J2ME:** MQe ist mit den J2ME-Technologien konform:

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

Genauer gesagt ist es mit deren gängigsten Profilen konform:

1. CDC/Foundation
2. CLDC/Midp

Bei der Entscheidung, welches von diesen verwendet werden soll, ist Folgendes zu beachten:

1. Bei Verwendung von CDC/Foundation wird der volle Funktionsumfang von MQe aktiviert, mit Ausnahme von Beispielcode, der das AWT-GUI-Paket erfordert.
2. Bei Verwendung von CLDC/Midp dagegen wird die Anwendung auf das rein clientseitige Verhalten sowie auf eine beschränkte Auswahl eingebauter Komprimierungs- und Verschlüsselungsfunktionen beschränkt.

**Anmerkung:** Geräte, bei denen CLDC/Midp aktiviert ist, können gravierende Speicherbeschränkungen haben, die die direkte Verwendung des Kernpakets von MQe verhindern.

**C-Bindings-Umgebung:** Die C-Bindings-APIs erfordern J2SE 1.3 oder später und können nur auf folgenden Plattformen installiert und verwendet werden:

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

**JMX-Schnittstelle:** Die Schnittstelle "Java Management Extensions Interface" (MQe JMX) wird als Anwendung betrieben, die in einer Java Virtual Machine (JVM) ausgeführt wird. Sie erfordert lediglich einen aktivierten lokalen WS-Manager. Wo diese Voraussetzung erfüllt ist, kann die Schnittstelle den instrumentierten lokaler WS-Manager und dessen Ressourcen verwalten. Sie kann auch über Remotezugriff aktivierte MQe-WS-Manager (und deren Ressourcen) verwalten, für die der lokale WS-Manager direkte Verbindung zum MQe-Netzwerk aufnehmen kann.

Die JMX-API von MQe ist in Übereinstimmung mit der JMX-Spezifikation V1.2 entwickelt worden und erfordert daher eine konforme Implementierung der JMX-Spezifikation. Wenn Sie nicht über eine solche Implementierung verfügen, können Sie auf eine JMX-Referenzimplementierung von SUN zurückgreifen: <http://java.sun.com/products/JavaManagement/>.

## Native C-Umgebung

Allgemeine Informationen finden Sie im Handbuch *C Programming Reference*, insbesondere auf der Seite *Compilation Information*. Die Seite ist allerdings nicht mehr ganz aktuell; eine Aktualisierung bietet das vorliegende Thema.

Die native C-Codebasis wird auf fünf Plattformen unterstützt:

- PocketPC2000
- PocketPC2002
- PocketPC2003
- WinCE .NET 4.2
- Windows 32 Bit

Auf WinCE .NET 4.2 werden die C-DLLs von WebSphere MQ Everywhere über die von der .NET-Umgebung angebotenen Platform Invocation Services unterstützt.

Für PocketPC werden Binärdateien sowohl für das Gerät als auch für den Emulator bereitgestellt, der in der integrierten Entwicklungsumgebung Microsoft Embedded Visual C++ verfügbar ist. Die für die Geräte bereitgestellten Binärdateien sind für ARM-Prozessoren kompiliert.

## Binärdateien

Das Stammverzeichnis der Binärdateien, der Dokumentation und der Beispiele ist das Verzeichnis C unterhalb des Verzeichnisses, das Sie für die Installation von MQe auswählen.

Unterhalb des Verzeichnisses C sind die Dateien dann wie folgt abgelegt:

### PocketPC2000

#### ARM

**DLLs** C:\PocketPc2000\arm\bin

**LIBs** C:\PocketPc2000\arm\lib

#### Emulator

**DLLs** C:\PocketPc2000\x86emulator\bin

**LIBs** C:\PocketPc2000\x86emulator\lib

## PocketPC2002

### ARM

**DLLs** C:\PocketPc2002\arm\bin

**LIBs** C:\PocketPc2002\arm\lib

### Emulator

**DLLs** C:\PocketPc2002\x86emulator\bin

**LIBs** C:\PocketPc2002\x86emulator\lib

## PocketPC2003 und WinCE .NET 4.2

### ARM

**DLLs** C:\PocketPc2003\arm\bin

**LIBs** C:\PocketPc2003\arm\lib

### Emulator

**DLLs** C:\PocketPc2003\x86emulator\bin

**LIBs** C:\PocketPc2003\x86emulator\lib

## Windows 32 Bit

**DLLs** C:\Win32\Native\bin

**LIBs** C:\Win32\Native\lib

## Headerdateien

Die Headerdateien sind bei allen nativen Plattformen die gleichen und sind im Verzeichnis include unter dem Installationsverzeichnis zu finden.

### MQe\_API.h

Dies ist die "Root"-Headerdatei. Wenn sie eingeschlossen wird, werden alle relevanten Headerdateien automatisch mit eingeschlossen.

Um sicherzustellen, dass die richtigen Dateien und Definitionen eingeschlossen werden, müssen Sie wie folgt angeben, dass Sie mit der nativen Codebasis arbeiten:

```
#define NATIVE // oder als Compileroption angeben
#include <published/MQe_API.h>
```

## Verlinken

Sorgen Sie für Verlinkung mit den folgenden zwei Bibliotheken:

### HMQ\_nativeAPI.lib

// die API-Bibliothek

### HMQ\_nativeCnst.lib

// die Bibliothek für statische konstante MQeString-Objekte

Im Allgemeinen sollten diese Dateien beide eingeschlossen werden. Ein optimierender Linker sollte dann Links zu nicht verwendeten Funktionen und Konstanten entfernen.

Die übrigen MQe-Bibliotheken werden statisch und dynamisch mit der Haupt-API-Bibliothek verlinkt und werden nach Bedarf mit eingeschlossen.



## Lizenzierung

MQe ist ein Toolkit, das es Benutzern ermöglicht, MQe-Anwendungen zu schreiben und eine Umgebung zu erstellen, in der diese ausgeführt werden können. Bevor Sie dieses Produkt oder Anwendungen, von denen das Produkt verwendet wird, einsetzen, müssen Sie sicherstellen, dass Sie über die erforderlichen Lizenzen verfügen.

1. Die Preise für Lizenzen für die Verwendung des Programms auf **Servern** beruhen auf **Processor License Units** (Prozessorlizenzeinheiten). Die Verwendung einer Kopie des Programms auf einem Server erfordert jeweils den Erwerb einer *Processor License Unit* pro Prozessor bzw. symmetrischen Multiprozessor in dem Server, auf dem die Kopie des Programms ausgeführt werden soll. Je nachdem, ob das Programm auf Point-of-Sale-Computern, d. h. Einzelhandelscomputern, oder Bauteilcomputern oder einer anderen Art von Computern ausgeführt wird, sind unterschiedliche Kategorien von *Processor License Units* und *Device Use Authorizations* (Gerätenutzungsberechtigungen) erforderlich. Die Verwendung des Programms auf Einzelhandelscomputern erfordert eine **Retail Server License** (Einzelhandelsserverlizenz); die Verwendung auf anderen Computern (Nicht-Einzelhandelscomputern) erfordert dagegen eine **Network Server License** (Netzwerkserverlizenz).
2. Für jede Verwendung des Programms auf einem separaten **Client-Gerät** ist eine zusätzliche *Device Use Authorization* erforderlich, außer bei den Geräten, die in der unter 3. beschriebenen *Network Server License* mit inbegriffen sind.
3. Jede *Network Server License* beinhaltet die Berechtigung zur eingeschränkten Verwendung des Programms auf nicht mehr als einhundert (100) Client-Geräten, unter der Bedingung, dass diese Kopien alle im selben Unternehmen bzw. in derselben Organisation verwendet werden wie die Server-Kopie.

Einzelheiten zu diesen Einschränkungen finden Sie unter <http://www.ibm.com/software/integration/wmqe/>.

Zur Verwendung des Produkts auf bestimmten Client-Plattformen (außer für Zwecke der Code-Entwicklung und des Testens) sind entsprechende *Device platform use authorizations* (Berechtigungen zur Nutzung auf Einheitenplattformen) erforderlich, die in den Berechtigungsnachweisen festgehalten sind und die für die Unterstützung der Verwendung von MQe gültig sind. Mit diesen Lizenzen ist der Benutzer jedoch nicht dazu berechtigt, die Brückenfunktion von MQe zu verwenden oder das Produkt auf den Serverplattformen einzusetzen, die in den Preisgruppenlisten von MQe enthalten sind. Diese Listen wurden von IBM veröffentlicht und stehen auch im Internet unter der nachstehenden URL zur Verfügung.

Einzelheiten zu diesen Einschränkungen finden Sie unter <http://www.ibm.com/software/integration/mqfamily/>.

---

## Installation von MQe

Dieses Kapitel stellt eine Anleitung zur Installation von MQe auf Maschinen dar, die für die Entwicklung von MQe-Anwendungen genutzt werden sollen.

Beim Installationsprogramm von MQe handelt es sich um eine Java-Datei vom Typ JAR mit plattform-spezifischen Ausführungsprogrammen, die direkt von der Produkt-CD aus ausgeführt werden können. Das Installationsprogramm extrahiert dabei die Arbeitsdateien in ein temporäres Verzeichnis, kopiert die MQe-Dateien auf den Computer und löscht anschließend die Arbeitsdateien.

Es ist zu beachten, dass in diesem Release der Anwendungs- und Lösungsanbieter für den Einsatz von MQe auf mobilen Endgeräten verantwortlich ist.

## Installationsvorgang

Die Informationen in diesem Abschnitt gelten für die Installation unter Windows, AIX, Linux, Solaris und HP-UX.

Während der Installation können Sie jederzeit auf die Schaltfläche **Zurück** des jeweiligen Bildschirms klicken, um zu vorangegangenen Bildschirmen zurückzukehren und Angaben nachzulesen oder zu ändern. Klicken Sie auf **Abbrechen** in einer der Anzeigen, wenn die Installation abgebrochen werden soll.

#### **Vor dem Start:**

- Es wird dringend geraten, eventuelle zuvor installierte Versionen von MQe vor der Installation bzw. Neuinstallation dieser neuen Version zu deinstallieren (siehe „Deinstallation von MQe“ auf Seite 12). Das Installationsprogramm erkennt Versionen des Produkts, die älter sind als Version 1.2.4, nicht und zeigt daher keine Warnungen an.
- Wenn Sie mit **Windows** arbeiten, achten Sie darauf, dass Ihre Benutzer-ID Administratorzugriff hat. Andernfalls kann es sein, dass die Startmenü-Symbole für IBM MQe nicht angezeigt werden.
- Wenn Sie mit **AIX** arbeiten, müssen Sie als Root-Benutzer angemeldet sein, um die Installation erfolgreich durchführen zu können.

#### **So installieren Sie MQe:**

1. Legen Sie die CD mit der Software in das CD-ROM-Laufwerk ein.
2. Starten Sie die Installation entweder von dem plattformspezifischen Ausführungsprogramm aus oder von der Datei 'setup.jar' aus:

- **Installation vom plattformspezifischen Ausführungsprogramm aus:**

- a. Die Ausführungsprogramme befinden sich in den plattformspezifischen Unterverzeichnissen auf der Produkt-CD. Rufen Sie zum Starten des Installationsprozesses das zu Ihrer Plattform passende Ausführungsprogramm auf (zum Beispiel setup.exe für Windows).
- b. Wenn Sie das Ausführungsprogramm auf Ihren lokalen Rechner kopieren, müssen Sie auch die Datei setup.jar in das Verzeichnis kopieren, das demjenigen übergeordnet ist, in das Sie das Ausführungsprogramm kopiert haben. Das Ausführungsprogramm kann nicht von einem Stammverzeichnis aus ausgeführt werden.
- c. Wenn Sie die Installation von einem Ausführungsprogramm aus aufrufen und ein Nachrichtenfenster mit dem Text No matching JVM was found (Keine passende JVM gefunden) angezeigt bekommen, war das Installationsprogramm nicht in der Lage, eine verwendbare Java-Umgebung zu finden. Wenn Sie diese Nachricht erhalten, müssen Sie zum Installieren die Datei setup.jar verwenden.
- d. Soll das Ausführungsprogramm eine bestimmte JVM verwenden, verwenden Sie die folgende Option:

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

Soll das Ausführungsprogramm ein bestimmtes Verzeichnis für temporäre Dateien verwenden, verwenden Sie die folgende Option:

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

- **Installation von der Datei setup.jar aus:**

Wechseln Sie in das Verzeichnis der Produkt-CD, in dem sich die Datei setup.jar befindet, und führen Sie mit dem Java-Befehl auf Ihrem Rechner das Installationsprogramm aus. Dieser Befehl lautet normalerweise java, jre oder jview. Beispiel:

#### **Unter Windows**

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

(Bei Verwendung von JVM 1.2.2 oder höher können Sie die jar-Datei über einen Doppelklick ausführen.)

#### **Unter Linux, AIX, Solaris und HP-UX**

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

- Bestätigen Sie auf dem Begrüßungsbildschirm die Installation von MQe, indem Sie auf **Weiter** klicken. Folgen Sie zum Abschließen der Installation den Eingabeaufforderungen.

## Unbeaufsichtigte Installation

Das Installationsprogramm kann im unbeaufsichtigten Modus ausgeführt werden. Dies bedeutet, dass während der Installation keine Fenster und Eingabeaufforderungen angezeigt werden. Es gibt zwei Möglichkeiten zur unbeaufsichtigten Installation:

### Von der jar-Datei aus

Führen Sie die Installation wie zuvor beschrieben aus, geben Sie jedoch die Option `-silent` an. Beispiel:

#### Unter Windows

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

#### Unter Linux, AIX, Solaris und HP-UX

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

### Von einem plattformspezifischen Ausführungsprogramm aus

Verwenden Sie die Option `-is:silent -silent`. Beispiel:

#### Unter Windows

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

#### Unter AIX

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

#### Unter Linux

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

#### Unter Solaris

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

#### Unter HP-UX

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

## Verzeichnisse bei der unbeaufsichtigten Installation

Standardmäßig wird MQe vom Installationsprogramm in den in untenstehender Tabelle aufgeführten Verzeichnissen installiert. Falls Sie irgendwelche früheren Versionen von MQe auf dem Rechner installiert haben, wird bei einer unbeaufsichtigten Installation Ihr aktuelles Verzeichnis verwendet, und nicht die Standardverzeichnisse aus dieser Tabelle:

Tabelle 1. Standardinstallationsverzeichnisse

Plattform	Standardinstallationsverzeichnis
iSeries	/opt/MQe
AIX	/opt/MQe
Linux	/opt/MQe
Solaris	/opt/MQe
Win32	\Programme\MQe
HP-UX	/opt/MQe

Wenn Sie bei einer unbeaufsichtigten Installation ein anderes Installationsverzeichnis angeben möchten, geben Sie die Option `-P MQe.installLocation` an:

### Von der jar-Datei aus

```
java run -silent -P MQe.installLocation="C:\Installationsverzeichnis"
```

### Von einem plattformsspezifischen Ausführungsprogramm aus

```
setup.exe -is:silent -silent -P MQe.installLocation="C:\Installationsverzeichnis"
```

**Anmerkung:** Wenn Sie bei einer unbeaufsichtigten Installation das Installationsverzeichnis mit dem Parameter `-P MQe.installLocation` ändern, müssen Sie den vollständigen Pfad zum neuen Ziel angeben.

## Unbeaufsichtigte Installation mit Optionsdateien

Bei einer unbeaufsichtigten Installation können Sie eine Optionsdatei angeben. Dies eröffnet folgende Möglichkeiten:

- Die Installation unbeaufsichtigt durchführen
- Das Installationsverzeichnis ändern
- Die zu installierenden Komponenten auswählen

Die folgende Beispiel-Optionsdatei setzt den Installationsmodus auf "unbeaufsichtigt", wählt als Installationsverzeichnis das Verzeichnis `C:\MQe` aus und legt fest, dass der komplette Funktionsumfang installiert wird.

```
#unbeaufsichtigte Installation durchführen
-silent
#Komponenten aktivieren
-P Java.active=true
-P Documentation.active=true
-P CBindings.active=true
-P Native.active=true
-P Palm.active=false
#Installationsverzeichnis festlegen
-P MQe.installLocation="C:\MQe"
```

### Anmerkung:

1. Nehmen Sie in die Optionsdatei das Flag `-is:silent` auf, wenn die Installation von einem Ausführungsprogramm aus erfolgen soll.
2. Die Datei `options.txt` darf keine Leerzeilen enthalten.
3. Beginnen Sie alle Zeilen mit dem Zeichen `#` oder einem zulässigen Befehl.
4. Eine Zeile kann mehrere Befehle enthalten.

Die folgenden Beispiele zeigen, wie das Installationsprogramm zusammen mit einer Optionsdatei verwendet wird:

### Von der JAR-Datei aus

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

### Von einem Ausführungsprogramm aus

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

## Befehlszeilenoption für den Eingabehilfenmodus

Die Befehlszeilenoption 'accessibility' für den Eingabehilfenmodus bewirkt, dass die Installation automatisch im Konsolenmodus erfolgt, und gibt darüber hinaus für Benutzer mit eingeschränktem Sehvermögen Informationen auf akustischem Wege aus. Beispiel für die Syntax:

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

Dieses Flag gilt auch bei der Deinstallation. Beispiel für die Syntax:

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

## Von einer ZIP-Datei aus installieren

Die MQe-Klassen stehen auch als zip-Datei zur Verfügung. Diese Datei kann für die Installation von MQe auf Einheiten verwendet werden, auf denen sich die Verwendung des grafischen Installationsprogramms nicht empfiehlt bzw. auf denen es nicht unterstützt wird. Auf UNIX-Systemen (z. B. Linux und HP-UX) müssen Sie einen Ordner erstellen, die entsprechende .zip-Datei dort hineinkopieren und dann die Klassendateien mit Hilfe eines Dekomprimierungsprogramms entpacken. Beispiel:

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

Nachdem die Klassendateien entpackt wurden, müssen Sie Ihre Umgebung für die Ausführung von MQe-Programmen konfigurieren.

## Installierte Komponenten

Bei der Installation von MQe können Sie verschiedene Funktionen auswählen. Diese Funktionen werden in den folgenden Themen beschrieben, zusammen mit den Komponenten, die bei der Auswahl einer Funktion installiert werden.

### MQe for Java

#### Java-Klassen von MQe

Eine Gruppe von Klassen, die den gesamten Funktionsumfang von MQe implementieren. Mit Untergruppen dieser Klassen können verschiedene MQe-Konfigurationen realisiert werden, z. B. Untergruppen für ein Gerät oder für einen Server.

#### Hilfsklassen

Eine Gruppe von Klassen, die von den Basisklassen abgeleitet wurden und einige der gängigsten Funktionen implementieren.

#### Beispielklassen

Eine Gruppe von Klassen, die die Verwendung der verschiedenen Funktionen von MQe veranschaulichen. Der Quellcode dieser Klassen steht ebenfalls zur Verfügung.

#### Dienstprogramme

Tools zur Unterstützung der Programmier- und Verwaltungsaufgaben in MQe.

## C-Bindings von MQe

### Spezielle Klassen für die C-Bindings

Dazu gehört das Paket `com.ibm.mqe.bindings`. Diese Klassen werden nur in Zusammenhang mit den C-Bindings benötigt. Sie haben keinen Einfluss auf die vorhandenen Funktionen.

### Header- und Binärdateien

Durch die C-Bindings werden Binärdateien und Header-Dateien hinzugefügt. Eine vollständige Beschreibung dieser Dateien finden Sie im Handbuch *MQe C Bindings Programming Guide*.

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

Dieses Handbuch enthält Anleitungen und Verfahrenshinweise für die Erstellung von C-Anwendungen für MQe und für die Systemadministration. Der Name der Datei lautet `hmq9a1_WMQE_C_BindingsProgrammingGuide.pdf`.

Informationen dazu finden Sie auch in der HTML-Version: [C Bindings Programming Reference](#).

## MQe for Native Platforms

### Header-Dateien

Alle nativen Plattformen erfordern dieselben Header-Dateien; diese werden gemeinsam mit den C-Bindings benutzt.

### Binärdateien

Es sind DLL- und LIB-Dateien, die mit den folgenden SDKs erstellt wurden, verfügbar:

- PocketPC 2002-SDK, ARM-Prozessor und PocketPC 2002-Emulator
- PocketPC 2003-SDK, ARM-Prozessor und PocketPC 2003-Emulator

### Beispiele

Dazu finden Sie auch Beispiele im Handbuch 'MQe Native Platforms Programming Reference'.

### PocketPC-Versionsinformationstool

Das Versionsinformationstool von MQe wird als Teil der nativen Codebasis ausgeliefert. Es handelt sich dabei um eine ausführbare Datei, die es Ihnen ermöglicht, die Versionsinformationen zu den installierten nativen DLL-Dateien zu überprüfen. Dieses Tool wird für PocketPC 2000, 2002 und 2003 erstellt.

Die Programmdatei liegt im Verzeichnis C:\tools\Version\PocketPc2000\arm bzw. im Verzeichnis C:\tools\Version\PocketPc2002\arm der Installation. Diese Datei muss an eine geeignete Stelle auf dem Gerät kopiert werden. Bei der Ausführung versucht das Tool alle nativen DLL-Dateien zu laden. In das Root-Verzeichnis des Gerätes wird eine Datei mit Informationen über die DLL-Dateien geschrieben.

- Wenn Sie das Tool ohne Argumente direkt aus dem Dateiverwaltungsprogramm bzw. von der Befehlszeile aus aufrufen, lädt das Tool die DLL-Dateien, die mit dem Standard-Ladepfad des Gerätes zu finden sind.
- Um ein Verzeichnis als Parameter anzugeben, rufen Sie das Tool auf der Befehlszeile auf. Die Angabe ist teilweise geräteabhängig, aber Sie haben die Möglichkeit, einen der für das PocketPC-Gerät verfügbaren Befehlszeilen-Interpreter zu verwenden. Wenn ein Verzeichnis angegeben wird, werden nur die in diesem Verzeichnis vorhandenen DLL-Dateien überprüft.

Zu jeder DLL-Datei gibt das Tool die Hauptversionsnummer an, einen möglichen EFix-Level, wobei es sich um die vierte Stelle der Versionsnummer handelt, und die ID des Builds, aus dem diese DLL-Datei stammt. Wenn sie bei manchen DLL-Dateien festgelegt haben, dass sie nicht auf das Gerät kopiert werden sollen, erhalten Sie Nachrichten dieses Inhalts.

**Anmerkung:** Wenn Sie DLL-Dateien aus verschiedenen Builds bzw. Versionen zu sehen bekommen, überprüfen Sie noch einmal, ob Sie tatsächlich die richtigen Dateien kopiert haben. In der Installation gibt es sowohl DLL-Dateien für die PocketPC-Emulatoren als auch für die Geräte. Achten Sie darauf, dass Sie die richtigen Dateien kopieren. Andernfalls kommt es beim Versuch, die Programme auszuführen, zu Fehlern.

### Dokumentation

Die bei früheren Versionen von MQe mitgelieferten Versionen sind grundlegend überarbeitet und zu diesem MQe-Information-Center umstrukturiert worden.

### Komponenten im Web

Im Regelfall besteht zur Entwicklung von Anwendungen Bedarf an der Installation von mehr als nur dem MQe-Produkt.

### Installation überprüfen

In diesem Abschnitt erfahren Sie, wie Sie durch Ausführung bestimmter Beispiele die erfolgreiche Installation eines MQe Development Kits überprüfen können.

## Java-Installation überprüfen

Nach der Installation von MQe können Sie wie folgt vorgehen, um durch Ausführung bestimmter Beispiele festzustellen, ob die Installation des Development Kits erfolgreich war.

- Stellen Sie sicher, dass die Java-Umgebung wie im Abschnitt „Java-Umgebung“ auf Seite 1 beschrieben eingerichtet ist. Bei der Ausführung einer der in diesem Abschnitt beschriebenen Windows-Stapeldateien ist jeweils der erste Parameter der Name des zu verwendenden Java Development Kits. Wenn Sie keinen Namen angeben, ist der Standard IBM.

**Anmerkung:** Die UNIX-Shellscripts haben keinen entsprechenden Parameter.

- Wechseln Sie in das richtige Verzeichnis:

### Windows

Wechseln Sie in das Verzeichnis <MQe-Installationsverzeichnis>\Java\demo\Windows.

**UNIX** Wechseln Sie in das Verzeichnis <MQe-Installationsverzeichnis>/Java/demo/UNIX.

- Erstellen Sie einen Warteschlangenmanager wie folgt:

### Windows

Führen Sie die Stapeldatei aus:

```
C example
CreateExampleQM.bat <JDK>
```

**UNIX** Führen Sie das Shellscript aus:

```
C example
CreateExampleQM
```

Damit erstellen Sie einen Beispiel-Warteschlangenmanager namens ExampleQM.

Im Zuge des Erstellungsvorgangs werden Verzeichnisse für Warteschlangenmanager-Konfigurationsinformationen und für Warteschlangen angelegt. Im Beispiel wird ein zum aktuellen Verzeichnis relatives Verzeichnis namens ExampleQM verwendet. Dieses Verzeichnis enthält zwei weitere Verzeichnisse:

- Registry - enthält Dateien, die Konfigurationsdaten von Warteschlangenmanagern aufnehmen.
  - Queues - für jede Warteschlange gibt es ein eigenes Unterverzeichnis, das die Nachrichten in dieser Warteschlange aufnimmt. (Dieses Verzeichnis wird erst angelegt, wenn die Warteschlange aktiviert wird.)
- Führen Sie eine einfache Anwendung wie folgt aus:

Nachdem Sie einen Warteschlangenmanager erstellt haben, können Sie ihn starten und in Anwendungen verwenden. Sie können die Stapeldatei ExamplesMQeClientTest.bat bzw. das Shellscript ExamplesMQeClientTest verwenden, um einige der Beispiele für einfache Anwendungen auszuführen.

Die Stapeldatei führt standardmäßig examples.application.Example1 aus. In diesem Beispiel wird eine Testnachricht an den Warteschlangenmanager ExampleQM geschickt und dann aus demselben Warteschlangenmanager empfangen. Wenn die zwei Nachrichten übereinstimmen, ist die Anwendung erfolgreich ausgeführt worden.

Im Paket examples.application finden Sie eine Reihe von Anwendungen, die verschiedene Leistungsmerkmale von MQe veranschaulichen. Diese Beispiele können Sie wie folgt ausführen:

### Windows

Übergabe der Parameter an die Stapeldatei:

```
C example
ExamplesMQeClientTest <JDK> <Beispielnummer>
```

**UNIX** Übergabe der Parameter an das Shell-Script:

```
C example
ExamplesMQeClientTest &gt;Beispielnummer&lt;
```

Dabei ist *Beispielnummer* das Suffix des jeweiligen Beispiels. Diese Nummer liegt zwischen 1 und 6 (je einschließlich).

- Löschen eines WS-Managers.

Wenn ein Warteschlangenmanager nicht mehr benötigt wird, können Sie ihn löschen. Zum Löschen des Beispielwarteschlangenmanagers Beispiel-WS:

#### Windows

Ausführung der Stapeldatei

C example

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

#### UNIX Ausführung des Shell-Scripts

C example

```
DeleteExampleQM
```

Wenn Sie einen Warteschlangenmanager gelöscht haben, können Sie ihn nicht mehr starten.

**Anmerkung:** In den Beispielen werden zur Erleichterung der Einrichtung relative Verzeichnisangaben verwendet. Es wird aber dringend empfohlen, außer bei elementarer Entwicklung und zu Demonstrationszwecken absolute Verzeichnisangaben zu verwenden. Wenn sich das aktuelle Arbeitsverzeichnis ändert und Sie mit relativen Verzeichnisangaben arbeiten, kann der Warteschlangenmanager seine Konfigurationsinformationen und Warteschlangen nicht mehr finden.

## C-Installation überprüfen

Nach der Installation von MQe können Sie einige Beispiele aus dem Handbuch "MQe C Programming Reference" ausführen, um Ihre Installation zu überprüfen.

---

## Installation modifizieren

Wie Sie Ihre Installation entfernen können, finden Sie im Abschnitt „Deinstallation von MQe“ erklärt. Beachten Sie, dass mit *Add/Remove Programs* (Programme hinzufügen/entfernen) keine installierten Optionen modifiziert werden können; es kann nur eine vollständige Deinstallation vorgenommen werden.

Wenn Sie Ihre Installation modifizieren möchten, z. B. indem Sie ein Leistungsmerkmal oder eine Komponente hinzufügen, die Sie bei der zuvor erfolgten Installation nicht ausgewählt hatten, legen Sie die Produkt-CD in Ihr Laufwerk ein, und folgen Sie der Anleitung im Abschnitt „Installationsvorgang“ auf Seite 5.

---

## Deinstallation von MQe

Lesen Sie die für Sie relevanten Anweisungen.

### Windows

Um MQe von einem Windows-System zu deinstallieren, haben Sie folgende Vorgehensweisen zur Auswahl:

#### Über die Systemsteuerung von Windows

1. Klicken Sie auf **Start->Einstellungen->Systemsteuerung**.
2. Klicken Sie doppelt auf das Programmsymbol **Add/Remove Programs (Programme hinzufügen/entfernen)**.
3. Klicken Sie im Dialogfenster *Add/Remove Programs* auf **IBM MQe**.
4. Klicken Sie auf **Hinzufügen/Entfernen...**, um das Deinstallationsprogramm zu starten.

Gehen Sie anhand der angezeigten Anweisungen vor, bis das Programm den Abschluss der Deinstallation meldet.



## Mit 'uninstall.exe'

Klicken Sie im Verzeichnis <MQe-Installationsverzeichnis> doppelt auf uninstall.exe, oder verwenden Sie folgenden Befehl:

```
<MQe-Installationsverzeichnis>\Uninst\uninstall.exe
```

Dabei ist <MQe-Installationsverzeichnis> das Verzeichnis, in dem Sie MQe installiert haben.

Folgen Sie der Bedienerführung, bis das Programm meldet, dass die Deinstallation abgeschlossen ist.

## Mit 'uninstall.jar'

Verwenden Sie die Datei uninstall.jar wie folgt:

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

## Unix

Folgen Sie dieser Anleitung zum Deinstallieren von MQe unter den folgenden Betriebssystemen:

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

Um MQe von Ihrem System zu deinstallieren, haben Sie folgende Vorgehensweisen zur Auswahl:

**Anmerkung:** Verwenden Sie unter AIX stets eine dieser Methoden, und *verwenden Sie nicht SMIT*, denn damit wird das Produkt nicht korrekt entfernt.

**Anmerkung:** Verwenden Sie unter Solaris stets eine dieser Methoden, und *verwenden Sie nicht pkgrm*, denn damit wird das Produkt nicht korrekt entfernt.

## Mit 'uninstall.bin'

Geben Sie folgenden Befehl ein:

```
<MQe-Installationsverzeichnis>/Uninst/uninstall.bin
```

<MQe-Installationsverzeichnis> ist das Verzeichnis, in dem Sie MQe installiert haben. Standardmäßig handelt es sich hierbei um /opt/MQe. Dieser Pfad wurde möglicherweise bei der Installation geändert.

Folgen Sie der Bedienerführung, bis das Programm meldet, dass die Deinstallation abgeschlossen ist.

## Mit 'uninstall.jar'

Verwenden Sie folgende Befehle, um die Datei uninstall.jar aufzurufen:

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

Folgen Sie der Bedienerführung, bis das Programm meldet, dass die Deinstallation abgeschlossen ist.

## Unbeaufsichtigte Deinstallation

Das Deinstallationsprogramm kann auch im unbeaufsichtigten Modus ausgeführt werden. Dies bedeutet, dass während der Deinstallation keine Fenster und Eingabeaufforderungen angezeigt werden. Es gibt zwei Möglichkeiten zur unbeaufsichtigten Deinstallation:

### Von der jar-Datei aus

Führen Sie die Deinstallation wie zuvor beschrieben aus, geben Sie jedoch die Option `-silent` an.  
Beispiel:

#### Unter Windows

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

#### Unter Linux, AIX, Solaris und HP-UX

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

### Von einem plattformspezifischen Ausführungsprogramm aus

Verwenden Sie die Optionen `-is:silent -silent`. Beispiel:

#### Unter Windows

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

#### Unter Linux, AIX, Solaris und HP-UX

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

## Unbeaufsichtigte Deinstallation mit Optionsdateien

Bei einer unbeaufsichtigten Deinstallation können Sie eine Optionsdatei angeben. Dies eröffnet folgende Möglichkeiten:

- Die Deinstallation unbeaufsichtigt durchführen
- Die zu deinstallierenden Komponenten auswählen

Im nachfolgenden Beispiel wird die Deinstallation unbeaufsichtigt ausgeführt. Dabei werden alle Komponenten mit Ausnahme der Dokumentation entfernt.

```
#unbeaufsichtigte Deinstallation durchführen
-silent
#Komponenten aktivieren
-P Java.active=true
-P Documentation.active=false-P CBindings.active=true
-P Native.active=true
```

### Anmerkung:

1. Nehmen Sie in die Optionsdatei das Flag `-is:silent` auf, wenn die Deinstallation von einem Ausführungsprogramm aus erfolgen soll.
2. Die Datei `options.txt` darf keine Leerzeilen enthalten.
3. Beginnen Sie alle Zeilen mit dem Zeichen `#...` oder einem zulässigen Befehl.
4. Eine Zeile kann mehrere Befehle enthalten.

Die folgenden Beispiele zeigen, wie das Deinstallationsprogramm zusammen mit einer Optionsdatei verwendet wird:

### Von der JAR-Datei aus

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

### Von einem Ausführungsprogramm aus

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

---

## Kapitel 2. Wartung von MQe

Wartungsaktualisierungen für MQe werden als eigenständiges neues Release geliefert.

Für ein Upgrade auf ein neues Release gibt es zwei Möglichkeiten:

**Deinstallieren Sie vollständig das bisherige Release, und installieren Sie das neue Release im selben Verzeichnis.**

Bewahren Sie das Installationspaket für das bisherige Release auf, für den Fall, dass Sie es später wiederherstellen möchten.

**Lassen Sie das vorhandene Release unangetastet, und installieren Sie das neue Release in einem neuen Verzeichnis.**

Überprüfen Sie nach der Installation die `classpath`-Angabe, um sicherzustellen, dass die richtige Version von MQe aufgerufen wird. Bei der Installation unter Windows müssen Sie noch sicherstellen, dass Sie dem Verknüpfungsordner für die neue Installation einen anderen Namen geben als dem für die bisherige.

Weitere allgemeine Informationen zu Wartungs-Updates und deren Verfügbarkeit finden Sie auf der Webseite der MQ-Produktfamilie unter folgender Adresse:  
<http://www.ibm.com/software/integration/mqfamily/>.

---

### Migration von 1.2.7 auf 2.0 oder 2.0.1

Falls Sie ein Upgrade auf Version 2.0 oder Version 2.0.1 durchführen, müssen Sie bedenken, wie sich die in diesem Abschnitt beschriebenen Änderungen auf Ihre MQe-Anwendung auswirken.

### Aliasnamen in MQeFields

In Version 1 ließ die an den MQe-WS-Manager übergebene MQeFields-Struktur die Angabe der folgenden zwei Aliasnamen zu:

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

Diese Aliasnamen gaben die Standardklassennamen an, die beim Laden von Attributsschlüsseln zu verwenden sind, falls keine Attributsschlüsselklasse angegeben ist.

**Diese Werte sind in der Codebasis von Version 2.0 und 2.0.1 fest codiert und können nicht mit Hilfe des Aliasmechanismus verändert werden. Angaben dieser Werte in INI-Dateien oder Aufrufen des MQe-WS-Managers werden ignoriert.**

### MQeFields

Für die Gewährleistung der Kompatibilität mit der Spezifikation "Java 2 Platform Micro Edition's (J2ME) Connected Limited Device Configuration (CLDC)/Mobile Information Device Protocol (MIDP)" wurden verschiedene Methoden aus MQeFields geändert bzw. entfernt:

- Die explizite Verwendung der Gleitkommatypen, `float` und `double`, wurde entfernt.

Beispielsweise könnten Sie `putFloat("Val1", -1.234)` verwendet haben.

Auf Java-Plattformen, die die Verwendung von `float/double` ermöglichen, kann diese Funktionalität dadurch nachgeahmt werden, dass die Daten mit der Methode `Java Object convert` der Basistypen explizit in die entsprechenden `int`- oder `long`-Werte konvertiert werden.

In diesem Fall wird der oben angeführte Methodenaufruf durch `putFloatAsInt("Val1", Float.floatToIntBits(-1.234))` ersetzt.

**Anmerkung:** Anwendungen aus Version 1 können diese Werte ganz normal abrufen.

- Die Methoden `dumpToFile` und `restoreFromFile` wurden entfernt.  
Anwendungen mit diesen Funktionen müssen nun das `MQeFields`-Objekt in einer Datei ablegen und die Bytefeldgruppe in die angegebene Datei schreiben.
- Die Verbindung abgelegter Daten durch exklusives Oder (XOR) ist ebenfalls nicht mehr möglich.

## **MQeChannel**

Die Klasse

`com.ibm.mqe.MQeChannel`

ist umbenannt worden und heißt nun:

`com.ibm.mqe.communications.MQeChannel`

Alle Verweise auf den alten Klassennamen in Verwaltungsnachrichten werden automatisch durch den neuen Klassennamen ersetzt.

## **MQeAttribute**

Im Zusammenhang mit 'MQeAttribute' wurden folgende Änderungen vorgenommen:

- Die Implementierung der Methode `equals()`

bei `MQeAttribute` und seinen Unterklassen in Version 1.2.7 (und früheren Versionen) ist in `isAcceptable()`

umbenannt worden.

- Mitgeliefert wird jetzt die Klasse 'MQeAttributeRule'.  
Attributregeln sollten dieser Klasse und nicht der Klasse 'MQeRule' entnommen werden.  
Alle Methoden von `MQeAttribute` und seinen Unterklassen, die bisher ein `MQeRule`-Objekt als einen ihrer Parameter genommen haben, nehmen nun stattdessen ein `MQeAttributeRule`-Objekt.

## **Nicht weiter unterstützte Methoden und Klassen**

Die hier aufgeführten Klassen sind nicht mehr Bestandteil des Produkts.

Es wird empfohlen, alle Anwendungen, die von diesen Klassen Gebrauch machen, dahingehend zu ändern, dass sie stattdessen die äquivalenten Funktionen von MQe Version 2.0 und Version 2.0.1 verwenden.

Damit vorhandene Anwendungen vor der Aktualisierung ausgeführt werden können, stellt MQe die JAR-Datei `MQeDeprecated.jar` bereit.

Die Datei `MQeDeprecated.jar` enthält folgende Klassen:

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

Weitere Informationen zum Ersetzen der erwähnten Klassen können Sie der Auflistung der einzelnen Klassen im Handbuch 'Java Programming Reference' entnehmen.

## Sicherheit

Folgende Änderungen wurden im Zusammenhang mit Sicherheitsaspekten vorgenommen:

1. Die Klassen MQeCL und MQeRandom sind durch die Klasse CL von cryptoLite ersetzt worden.
2. Von Version 2.0 an entfällt die frühere Unterstützung von Minizertifikaten.

Diese Änderungen haben folgende Auswirkungen:

1. Die Klasse CL wird in Form der Datei 'cryptoLite.zip' geliefert. Zur Verwendung der Sicherheitsfunktionen von MQe muss diese ZIP-Datei im Java-Klassenpfad abgelegt werden.
2. 'MQeMiniCertificateServer' unterstützt die früher verwendeten Minizertifikate nicht mehr.

## Übertragungseinstellungen

In Version 2 ist das Standardzeitlimit von einer Stunde auf fünf Minuten reduziert worden. Außerdem sind die zur Festlegung bestimmter Adapter-Variablen (z. B. Socket-Zeitlimits) verwendeten Systemeigenschaften geändert worden.

Dies kann Auswirkungen auf Clients in langsamen oder instabilen Netzwerken haben.



---

## Index

### B

Betriebssysteme, unterstützte 1

### E

Erforderliche Betriebssysteme 1

### I

Installation

    Hinweise zur Migration 5

Installation überprüfen

    C 10, 11

    Java 10, 11

### M

Migration 5

### S

Softwareumgebungen 1

### U

Umgebungen, Software 1

Unterstützte Betriebssysteme 1

Upgrades 15

### V

Voraussetzungen 1, 5

### W

Wartung 15

Wartungs-Updates installieren 15