

WebSphere MQ Everyplace V2.0.2

Inhaltsverzeichnis

Willkommen bei MQe	1	Benötigte Hardware und Software.	28
Kurzüberblick über MQe	1	Zu verwendende Codebasis	28
Codebasis	3	MQe-Entwicklungszyklus.	29
Neuerungen in Version 2.0.2	5	Einstieg in MQe	30
Verwendung dieses Information Centers	5	Verwendung von MQe mit MQ.	30
Was ist MQe?	7	Weitere Informationen.	33
Einführung in MQe	7	Bemerkungen und Marken	34
Stellung von MQe in der MQ-Familie	7	Bemerkungen.	34
Mögliche Einsatzgebiete von MQe	9	Marken.	35
Wie MQe funktioniert	9	Glossar	36
MQe-SupportPacs	25		
Planung der Implementierung	27	Index	43
Lizenzen	27		

Willkommen bei MQe

Der vollständige Name dieses Produkts lautet WebSphere MQ Everyplace Version 2 Release 0 Modification 1 bzw. in der üblicheren Ausdrucksweise WebSphere MQ Everyplace V2.0.1.

In der vorliegenden Dokumentation wird das Produkt im Allgemeinen kurz als *MQe* bezeichnet.

MQe dient zum geschützten Nachrichtenaustausch bei Kleingeräten wie Sensoren, Telefonen und Personal Digital Assistants (PDAs) sowie bei Laptops und Desktop-Computern.

Programmierschnittstellen

Die Anwendungsprogrammierschnittstelle zu MQe wird in der vorliegenden Dokumentation als *MQe-API* bezeichnet. Die MQe-API unterstützt die Programmiersprachen Java und C wie folgt:

Java Bietet Zugriff auf alle MQe-Funktionen in Version 2.

Es gibt drei Versionen der C-Unterstützung:

Die native C-Codebasis

Bietet Zugriff auf einen Teil der MQe-Funktionen, wobei die Haupteinschränkung darin besteht, dass nur Geräte-Warteschlangenmanager verwendet werden können, so dass Nachrichten nur gesendet, aber nicht empfangen werden können.

Die C-Bindings

Werden für die Verwendung bis zur Verfügbarkeit des vollen Funktionsumfangs in der nativen C-Codebasis bereitgestellt. Ihre Anwendung ruft die C-API in den Bindings auf, und der Aufruf wird zur Ausführung durch MQe an die Java-Klassen weitergeleitet. Die Bindings wurden für Version 1 von MQe geschrieben, bieten aber auch in Version 2 von MQe noch Zugriff auf beinahe alle Java-Funktionen.

Weitere Einzelheiten zur Codebasis finden Sie im Abschnitt „Codebasis“ auf Seite 3.

Ferner finden Sie relevante Informationen im Abschnitt „Marken“ auf Seite 35.

Kurzüberblick über MQe

Wozu dient MQe?

- Zum geschützten Nachrichtenaustausch bei Kleingeräten wie Sensoren, Telefonen und Personal Digital Assistants (PDAs) sowie bei Laptops und Desktop-Computern.

Was ist unter Nachrichtenaustausch zu verstehen?

- Software (wie sie in MQ und MQe enthalten ist), die für Sie das Senden und Empfangen von Daten zwischen Anwendungen und über Netzwerke erledigt. Die Nachrichtenübermittlung ist *garantiert* und *entkoppelt* von der Anwendung; Anwendungsprogrammierer benötigen keine Kenntnis der Einzelheiten der Übertragungsprogrammierung.
- Wenn eine Anwendung Daten an eine andere Anwendung zu schicken hat, packt sie die Daten in Nachrichten und reiht die Nachrichten in eine *Warteschlange* ein.
- Die *Warteschlange* gehört einem *Warteschlangenmanager*, und von diesem wird sie betrieben.
- Eine weitere Anwendung (oder auch ein anderer Teil derselben Anwendung) kann so konfiguriert werden, dass sie sich auf eine der folgenden Weisen verhält:
 - Die betreffende Anwendung kann diese Nachrichten aus derselben Warteschlange entnehmen.

- Der Warteschlangenmanager kann so konfiguriert werden, dass er die Nachrichten in der Warteschlange über eine *Verbindung* über das Netzwerk an eine Warteschlange eines *fernen* Warteschlangenmanagers auf einem anderen Rechner schickt, aus der sie eine andere Anwendung dann entnimmt.
- Die Zielanwendung kann die Nachrichten nach Bedarf von sich aus über das Netzwerk holen.
- Einem Warteschlangenmanager können mehrere Warteschlangen gehören.
- MQe lässt nur einen Warteschlangenmanager pro JVM bzw. Prozess zu.

Was ist MQe?

- Ein Toolkit, das auf einer Reihe von Plattformen unterstützt wird:
- Die API ist in Java oder C verfügbar.
- Die Produktfunktion wird in Form von Java-Klassen und C-DLLs bereitgestellt.

Wie funktioniert es?

- Zunächst wird ein Warteschlangenmanager erstellt. Dabei handelt es sich um einen Satz von Angaben, der die anfängliche Grundkonfiguration des Warteschlangenmanagers beschreibt.
- Diese Angaben werden in der *MQe-Registry* gespeichert (Hinweis: Bei Windows-Systemen ist diese von der Windows-Registry zu unterscheiden).
- Jetzt kann auf Ihrem Gerät eine Anwendung diesen Warteschlangenmanager starten. Er wird so lange ausgeführt wie die Anwendung. Wenn die Anwendung gestoppt wird, wird auch der Warteschlangenmanager gestoppt.
- Ein Warteschlangenmanager kann während der Ausführung weitgehend konfiguriert und umkonfiguriert werden, indem *MQe-Verwaltungsnachrichten* an ihn gesendet werden (die zugleich die Informationen in der Registry aktualisieren). Im Normalfall generieren Sie diese Nachrichten mit den Verwaltungstools *MQe_Script* und *MQe_Explorer*.
- Auf Ihrem Server können Sie eine Anwendung schreiben, die den Warteschlangenmanager betreibt. Diese Anwendung kann dann dauerhaft ausgeführt werden, so dass Client-Warteschlangenmanager Nachrichten nach Bedarf senden können. Auf einem Windows-System kann eine Warteschlangenmanager-Anwendung als *Service* ausgeführt werden, so dass sie jedes Mal ausgeführt wird, wenn der Rechner gestartet wird.

Wie wird es verwendet?

- Um MQe zu verwenden, müssen Sie eine eigene Anwendung schreiben.
- Zum Erstellen Ihrer *MQe-Geräte-Warteschlangenmanager*-Konfiguration und zum Entwickeln Ihrer MQe-Anwendung verwenden Sie einen PC.
- Sie laden nur die Dateien und Komponenten, die Sie benötigen (für Ihre Anwendung ebenso wie für MQe), auf Ihr Gerät herunter und führen sie dort aus.
- Sie führen auf einem Server-Rechner einen *Server-Warteschlangenmanager* aus, mit dem die Geräte Verbindung aufnehmen und an den sie Nachrichten senden sollen.
- Für die Kommunikation mit MQ führen Sie (auf einem Server-Rechner) einen *Gatewaywarteschlangenmanager* aus, der als Zwischenglied agiert - er kann MQe-Nachrichten von Ihren Geräten (oder Ihrem Server) empfangen, sie in MQ-kompatible Nachrichten *umwandeln* und sie dann mit MQ austauschen.
- Sie können Warteschlangenmanager erstellen und Ihr MQe-Netzwerk aus der Ferne verwalten, indem Sie zwei herunterladbare SupportPacs verwenden: *MQe_Script* (in Scripten verwendbare Befehle) oder *MQe_Explorer* (GUI).
- Mit denselben Tools können Sie mit MQe experimentieren, ohne eine Anwendung zu schreiben.

Sind besondere Leistungsmerkmale vorhanden?

- Sie können *Adapter* nutzen, um MQe Geräteschnittstellen zuzuordnen. Beispiel:
 - *Kanäle* (an jedem Ende einer Verbindung) nutzen *Protokolladapter*, um HTTP, natives TCP/IP, UDP und andere Protokolle für die Übertragung einzusetzen.

- Warteschlangen nutzen *Feldspeicheradapter* als Schnittstelle zu einem Speichersubsystem wie etwa dem Speicher oder dem Dateisystem.
- Sie können auf drei Ebenen Sicherheitsfunktionen aktivieren:
 - Sicherheit auf lokaler Ebene**
Schützt nachrichtenbezogene Daten auf lokaler Ebene.
 - Sicherheit auf Nachrichtenebene**
Schützt Nachrichten zwischen der Ursprungsanwendung und der empfangenden Anwendung von MQe.
 - Sicherheit auf Warteschlangenebene**
Schützt Nachrichten zwischen dem Ursprungs-WS-Manager und der Zielwarteschlange.
- Sie können *Regeln* verwenden, um das Verhalten einiger der Hauptkomponenten von MQe anzupassen.
- Sie können Traceerstellung und Ereignisprotokollierung verwenden, um Fehler in Ihrer Anwendung leichter beheben zu können.

Weitere Informationen:

„Codebasis“

„MQe-SupportPacs“ auf Seite 25

Codebasis

Überblick

Die Anwendungsprogrammierschnittstelle von MQe (API: Application Programming Interface) ist die Programmierschnittstelle zu MQe. Die API von MQe unterstützt die Programmiersprachen Java und C wie folgt:

Die Java-Version

Diese bietet Zugriff auf alle MQe-Funktionen in Version 2. Die Klassen, Methoden und Prozeduren im Einzelnen sind in dem Dokument Java Programming Reference beschrieben. An zahlreichen Stellen in diesem Information Center finden sich Beispiele zur MQe-Programmierung.

Es gibt drei Versionen der C-Unterstützung:

Die native C-Codebasis

Diese bietet Zugriff auf einen Teil der MQe-Funktionen, wobei die Haupteinschränkung darin besteht, dass nur Geräte-Warteschlangenmanager verwendet werden können. Ferner gelten folgende Einschränkungen (siehe auch untenstehende Tabelle):

- Keine Unterstützung von Warteschlangen zum Speichern und Weiterleiten sowie von Warteschlangen mit Brückenfunktion
- Unterstützt nur den HTTP-Adapter
- Unterstützt nur die RLE-Komprimierungsfunktion
- Unterstützt nur die RC4-Verschlüsselungsfunktion
- Unterstützt nur das Attribut *MAttribute* und die lokalen Sicherheitseinrichtungen

Die Methoden und Prozeduren im Einzelnen sind im Dokument C Programming Reference beschrieben. An zahlreichen Stellen in diesem Information Center finden sich Beispiele zur MQe-Programmierung für die C-API.

Die C-Bindings

Diese werden für die Verwendung bis zur Verfügbarkeit des vollen Funktionsumfangs in der nativen C-Codebasis bereitgestellt. Ihre Anwendung ruft die C-API in den Bindings auf, und der Aufruf wird zur Ausführung durch MQe an die Java-Klassen weitergeleitet.

Die Bindings wurden für Version 1 von MQe geschrieben, bieten aber auch in Version 2 von MQe noch Zugriff auf beinahe alle Java-Funktionen. Die Methoden und Prozeduren im Einzelnen sind im Dokument C Programming Reference beschrieben. Beispiele zur MQe-Programmierung für die C-Bindings finden sich im Dokument C Bindings Programming Reference.

Arten von Warteschlangenmanagern

In der gesamten Dokumentation werden, wie in untenstehender Tabelle, die folgenden Warteschlangenmanagerbeschreibungen verwendet, die unbedingt zu unterscheiden sind:

Geräte-Warteschlangenmanager

Ein Warteschlangenmanager ohne Empfangsfunktionskomponente und ohne Brückenkomponente. Er kann daher lediglich Nachrichten senden, aber keine empfangen.

Server-Warteschlangenmanager

Ein Warteschlangenmanager, zu dem eine Empfangsfunktion hinzugefügt werden kann. Mit der Empfangsfunktion kann er auch Nachrichten empfangen, nicht nur senden.

Gateway-Warteschlangenmanager

Ein Warteschlangenmanager, zu dem eine Empfangsfunktion und eine Brücke hinzugefügt werden können. Mit der Empfangsfunktion kann er auch Nachrichten empfangen, nicht nur senden, und mit der Brücke kann er mit MQ kommunizieren.

Optionentabelle

Option	Java	Natives C	C-Bindungen
Betriebssysteme	Alle mit Java 2 (das mit Java Version 1.2 begann)	PocketPC2002; PocketPC2003; Windows (ab MQe V2.0.1.5)	Windows 32 Bit
WS-Manager	Alle Arten	Nur Geräte-WS-Manager	Alle Arten
Gateway zu MQ (WS-Manager mit Brücke und Empfangsfunktion)	Ja	Nein	Ja
Warteschlangen zum Speichern und Weiterleiten, Warteschlangen mit Brückenfunktion	Ja	Nein	Ja
Adapter	Alle	Nur HTTP-Adapter	Alle
Komprimierungsfunktionen	Alle	Nur RLE	Alle
Verschlüsselungsfunktionen	Alle	RC4	Alle
Sicherheitsfunktionen	Alle	Nur <i>MAttribute</i> und lokale Attribute	Alle
Nachrichten zu Traces hinzufügen	Ja	Nein	Ja
Ereignisprotokollierung	Ja	Nein	Ja
Private Registry und Berechtigungen	Ja	Nein	Ja
Attributregeln	Ja	Nein	Ja
Brückenregeln	Ja	Nein	Ja
Anpassbare Klassen	Alle	Nein	Einige
Anwendungen laden	Ja	Nein	Ja

Neuerungen in Version 2.0.2

Änderungen im Release WebSphere MQ Everyplace Version 2.0.2

- Die gesamte Dokumentation ist aus der Toolkit-Installation herausgenommen und ins Web gestellt worden.
- Die native C-Codebasis ist um UTF-16 mit Unterstützung von Ersatzzeichen erweitert worden.
- Unterstützung von WinCE .NET 4.2 in der nativen C-Codebasis. Es ist zu beachten, dass auf WinCE .NET 4.2 die C-DLLs von WebSphere MQ Everyplace über die von der .NET-Umgebung angebotenen Platform Invocation Services unterstützt werden.
- Es ist ein Abschnitt über Migration hinzugekommen. Dieser Abschnitt enthält Informationen über die Verwendung der neuen Klassen von MQe, die es Benutzern ermöglichen, Nachrichten an den WebSphere Message Broker über einen Gateway-Warteschlangenmanager zu senden, zu veröffentlichen und zu subscribieren.

Verwendung dieses Information Centers

Eingabehilfen

Die folgenden Abschnitte geben spezielle Hinweise zur Verwendung von Eclipse für die MQe-Dokumentation.

Allgemeinere Informationen zu Eingabehilfen in Eclipse finden Sie in der Hilfe von Eclipse selbst unter *Accessibility in the help system* (Eingabehilfen im Hilfesystem).

Tastaturhinweise

- Um zwischen Teilfenstern hin- und herzuwechseln, können Sie die Funktionstaste F6 verwenden, oder Strg-F6, falls Sie mit einem Browser arbeiten, der auf Mozilla 1.2 oder einer neueren Version von Mozilla beruht.
- In jedem Teilfenster und jeder Funktionsleiste können Sie die Tabulatortaste verwenden, um zwischen Symbolen oder Objekten hin- und herzuwechseln.
- Mit der Rückschritt-Taste gelangen Sie zurück zum zuvor angezeigten Thema. Mit der ALT-Taste in Verbindung mit der Rechtspfeil- bzw. Linkspfeiltaste können Sie vor- bzw. zurückblättern.
- Wenn Sie zu einem Thema über einen Link oder über eine Suche gelangt sind, können Sie herausfinden, wo es im Inhaltsverzeichnis eingeordnet ist, indem Sie auf das Symbol "Show in Table of Contents" (Im Inhaltsverzeichnis hervorheben) in der Themen-Funktionsleiste klicken. (Mit der Tastatur erreichen Sie das wie folgt: Verwenden Sie die Funktionstaste F6 oder - falls Sie mit einem Browser arbeiten, der auf Mozilla 1.2 oder einer neueren Version von Mozilla beruht - Strg-F6, um die Teilfenster bzw. Leisten zu durchlaufen, und verwenden Sie anschließend die Tabulatortaste, um in der Funktionsleiste das Symbol auszuwählen.)
- Zum Öffnen von Kontextmenüs können Sie die Umschalttaste in Verbindung mit F10 verwenden, oder die spezielle Kontextmenütaste, falls Ihre Tastatur über eine solche verfügt.

Die Teilfenster

- Das Navigationsteilfenster befindet sich auf der linken Seite. Es weist folgende Registerkarten auf:
 - Inhalt
 - Suchergebnisse
- Das Thementeilfenster ist dasjenige, in dem Sie jetzt gerade diese Worte lesen; darin wird jeweils das aktuelle Hilfethema angezeigt. Klicken Sie, wenn Sie direkt zu einem anderen Thema wechseln möchten, auf den entsprechenden hervorgehobenen Link (wenn Sie stattdessen die Tastatur verwenden möchten, verschieben Sie zunächst die Hervorhebung mit der Tabulatortaste bis zum gewünschten Link, und drücken Sie die Eingabetaste, um dem hervorgehobenen Link zu folgen). Klicken Sie zum Suchen von Wörtern auf der Seite in das Thema, und drücken Sie **Strg-F** (wenn Sie stattdessen die Tastatur verwenden möchten, verwenden Sie die Funktionstaste F6, um zwischen Teilfenstern hin- und

herzuwechseln, bzw. Strg-F6, falls Sie mit einem Browser arbeiten, der auf Mozilla 1.2 oder einer neueren Version von Mozilla beruht). Wenn Sie mit der Tastatur arbeiten möchten, verwenden Sie die Taste mit dem Aufwärtspfeil bzw. dem Abwärtspfeil, um die Seite nach oben bzw. unten zu verschieben.

- Die Funktionsleiste entlang dem oberen Rand dieses Thementeilfensters weist Symbole für folgende Funktionen auf: 'Vorwärts', 'Zurück', Show in Table of Contents (Im Inhaltsverzeichnis hervorheben) (nützlich nach einer Suche), und 'Drucken' (wenn Sie stattdessen die Tastatur verwenden möchten, verwenden Sie ALT-Linkspfeil bzw. ALT-Rechtspfeil und das Funktionsleistensymbol bzw. zum Drucken Strg-P; verwenden Sie die Funktionstaste F6, um zwischen Teilfenstern hin- und herzuwechseln, bzw. Strg-F6, falls Sie mit einem Browser arbeiten, der auf Mozilla 1.2 oder einer neueren Version von Mozilla beruht, und verwenden Sie anschließend auf der Funktionsleiste die Tabulatortaste, um das Symbol auszuwählen).
- Die Funktionsleiste über dem Navigationsteilfenster bietet Suchmöglichkeiten an.

Nützliche Zusatzfunktionen

- Wenn Sie zu einem Thema über die Suchfunktionsleiste gelangt sind, können Sie herausfinden, wo es im Inhaltsverzeichnis eingeordnet ist, indem Sie auf das Symbol Show in Table of Contents (Im Inhaltsverzeichnis hervorheben) in der Themen-Funktionsleiste klicken. (Mit der Tastatur erreichen Sie das wie folgt: Verwenden Sie die Funktionstaste F6 oder - falls Sie mit einem Browser arbeiten, der auf Mozilla 1.2 oder einer neueren Version von Mozilla beruht - Strg-F6, um die Teilfenster bzw. Leisten zu durchlaufen, und verwenden Sie anschließend die Tabulatortaste, um in der Funktionsleiste das Symbol auszuwählen.)
- Wenn Sie Wörter auf der Seite suchen möchten, klicken Sie in das Thementeilfenster (wenn Sie stattdessen die Tastatur verwenden möchten, verwenden Sie die Funktionstaste F6, um zwischen Teilfenstern hin- und herzuwechseln, bzw. Strg-F6, falls Sie mit einem Browser arbeiten, der auf Mozilla 1.2 oder einer neueren Version von Mozilla beruht), und drücken Sie anschließend **Strg-F**.
- Klicken Sie mit der rechten Maustaste auf einen Eintrag des Inhaltsverzeichnisses, und wählen Sie **Eigenschaften** aus, um ein Dialogfenster zu öffnen, aus dem Sie sich eine URL zur Verwendung bei künftigen Verweisen auf das Thema kopieren können; diese kann z. B. zur Lesezeichenliste Ihres Browsers hinzugefügt oder in eine E-Mail kopiert werden.

Drucken

- Klicken Sie auf das Drucksymbol in der Funktionsleiste, oder klicken Sie in das Thema, und drücken Sie Strg-P (wenn Sie stattdessen die Tastatur verwenden möchten, verwenden Sie die Funktionstaste F6, um zwischen Teilfenstern hin- und herzuwechseln, bzw. Strg-F6, falls Sie mit einem Browser arbeiten, der auf Mozilla 1.2 oder einer neueren Version von Mozilla beruht).

Schriftgröße

- Verwenden Sie zum Einstellen der Schriftgröße die entsprechenden browsereigenen Funktionen.

Hinweise zur Suchfunktion

Wenn Sie Themen in der gesamten Bibliothek suchen möchten (Find topics in the whole library), verwenden Sie die Suchfunktionsleiste unmittelbar über dem Navigationsteilfenster. Dabei ist Folgendes zu beachten:

- Verwenden Sie zum Einschränken des Suchbereichs in der Bibliothek die erweiterte Suche.
- Die Suche nach einer Zahl funktioniert bei einigen Versionen des Eclipse-Hilfe-Browsers nur, wenn Sie die Zahl in doppelte Anführungszeichen einschließen, z. B.: "2.0.1".
- Verwenden Sie für unvollständige Wörter das Zeichen *, z. B.: Mit **Admin*** werden Vorkommen von *Administration*, *Administrator* usw. gefunden.
- Für eine Suche nach mehreren Wörtern sind diese durch Leerzeichen zu trennen, z. B.: Mit **Warteschlangen verwalten** werden gleichermaßen die Themen mit dem Wort *Warteschlangen* und die Themen mit dem Wort *verwalten* gefunden.

- Für eine Suche nach einer Wortfolge ist diese in doppelte Anführungszeichen zu setzen, z. B.: Mit "**Warteschlangen verwalten**" werden nur die Themen gefunden, in denen die Wortfolge *Warteschlangen verwalten* als Ganzes vorkommt.
- Die gefundenen Wörter werden bei der Anzeige des Themas hervorgehoben.
- Um Wörter in einem angezeigten Thema zu finden, klicken Sie in das Thementeilfenster hinein (wenn Sie stattdessen die Tastatur verwenden möchten, verwenden Sie die Funktionstaste F6, um zwischen Teilfenstern hin- und herzuwechseln, bzw. Strg-F6, falls Sie mit einem Browser arbeiten, der auf Mozilla 1.2 oder einer neueren Version von Mozilla beruht), und drücken Sie **Strg-F**.

Was ist MQe?

Einführung in MQe

MQe ist Teil der WebSphere MQ-Familie von Nachrichtenaustauschprodukten zu Geschäftszwecken sowie der WebSphere Everyplace-Familie. Es dient zum Schreiben von eigenen Anwendungen, die *Nachrichten* austauschen, in die Daten gepackt sind, und sorgt für garantierte Nachrichtenübermittlung. MQe ist so angelegt, dass es sich gut mit anderen Teilen der WebSphere MQ-Familie integrieren lässt.

MQe ist so angelegt, dass es den Nachrichtenaustauschanforderungen bei Kleingeräten wie Sensoren, Telefonen und Personal Digital Assistants (PDAs) sowie bei Laptops und Desktop-Computern gerecht wird. Es unterstützt mobile Umgebungen und ist für die Verwendung in öffentlichen Netzen geeignet, da es den Anforderungen instabiler Kommunikationsnetze gerecht wird. Da viele MQe-Anwendungen außerhalb des durch eine Internet-Firewall geschützten Bereichs ausgeführt werden, bietet MQe auch Sicherheitsfunktionen an.

Zum Verständnis dieses Produktes und der Dokumentation ist ein Verständnis der Konzepte geschützter Nachrichtenübermittlung von Vorteil.

Wenn Sie sich damit noch nicht auskennen, kann folgendes WebSphere MQ-Handbuch für Sie von Nutzen sein: *An Introduction to Messaging and Queuing*, GC33-0805. Dieses Handbuch kann als Softcopy unter "Books" aus der WebSphere-MQ-Bibliothek im Internet heruntergeladen werden. Diese finden Sie über die Webseite von WebSphere MQ: <http://www.ibm.com/software/integration/websphere/library/>.

Stellung von MQe in der MQ-Familie

Elementarer Nachrichtenaustausch

Der Nachrichtenaustausch basiert bei allen Produkten bzw. Produktgruppen auf *Warteschlangen* und *Warteschlangenmanagern* (WS-Managern). Die WS-Manager verwalten Warteschlangen, in denen Nachrichten gespeichert werden können. Anwendungen kommunizieren jeweils mit einem *lokalen WS-Manager*, über den sie Nachrichten in Warteschlangen *einreihen* (PUT) bzw. aus diesen *entnehmen* (GET). Nachrichten, die in eine ferne, einem fernen WS-Manager zugeordnete Warteschlange eingereicht werden, werden über *Verbindungen* an diesen *fernen WS-Manager* übertragen. Auf diese Weise erfolgt die Übertragung von Nachrichten an das eigentliche Ziel unter Umständen über mehrere zwischengeschaltete WS-Manager. Das Wesentliche am Nachrichtenaustausch ist die Entkopplung der sendenden und der empfangenden Anwendung, indem Nachrichten bei Bedarf an zwischengeschalteten Punkten in Warteschlangen eingereicht werden.

MQ und MQe erbringen die Nachrichtenaustauschleistungen der MQ-Produktfamilie. Beide Produkte sind so angelegt, dass sie eine oder mehrere Hardwareserverplattformen und die meisten der zugehörigen Betriebssysteme unterstützen. Da die einzelnen Plattformen ein unterschiedliches Leistungsspektrum aufweisen, wurden die einzelnen Produkte in Produktgruppen zusammengefasst, denen jeweils ein gemeinsames Konzept in Funktion und Design zu Grunde liegt.

Verteilte Nachrichtenübertragung

WebSphere MQ for Windows NT, Windows 2000, AIX, iSeries, HP-UX, Solaris und andere Plattformen

Hostnachrichtenübertragung

WebSphere MQ for z/OS

Zeitnaher und standortunabhängiger Nachrichtenaustausch (Pervasive Messaging)

MQe for Windows, AIX, Solaris, Linux und HP-UX

Weitere Einzelheiten finden Sie im Abschnitt „Wie MQe funktioniert“ auf Seite 9.

MQe

MQe unterstützt eine Vielzahl von Netzkonfigurationen. Im Gegensatz zu den Hostprodukten oder den verteilten Produkten von MQ gibt es keine starre Einteilung in Clients und Server. Sie können vielmehr MQe-Warteschlangenmanager so konfigurieren, dass sie bald als Client, bald als Server agieren und somit anwendungsdefinierte Tasks ausführen können.

Ein Beispiel für speziell zugeschnittene Konfiguration ist, dass Sie MQe in die Lage versetzen können, Nachrichten mit MQ-Host-Warteschlangenmanagern auszutauschen. Dazu muss ein MQe-WS-Manager mit *Brückenfunktion* konfiguriert werden. Ohne die Brückenfunktion kann ein MQe-Warteschlangenmanager nur mit anderen MQe-Warteschlangenmanagern direkt kommunizieren. Eine indirekte Kommunikation über zwischengeschaltete WS-Manager mit Brückenfunktion im Netzwerk ist jedoch auch dann möglich.

Weitere Einzelheiten finden Sie im Abschnitt „Wie MQe funktioniert“ auf Seite 9.

Erweiterung der MQ-Produktfamilie durch MQe

MQe erweitert die Nachrichtenaustauschmöglichkeiten der MQ-Familie um folgende Merkmale:

- Es unterstützt Kleingeräte wie PDAs, Telefone und Sensoren. MQe unterstützt außerdem Zwischeneinheiten wie Laptops, Workstations, verteilte Systeme und Hostplattformen. MQe bietet gesicherte einmalige Nachrichtenzustellung und ermöglicht den Nachrichtenaustausch mit anderen Produkten der Produktfamilie.
- Es stellt Nachrichtenaustausch-Funktionen zur Verfügung, die wenig Ressourcen belegen.
- Es stellt umfassende Sicherheitseinrichtungen zum Schutz von Warteschlangen sowie von Nachrichten und zugehörigen Daten (während diese gespeichert sind bzw. übertragen werden) zur Verfügung.
- Es ermöglicht einen effizienten Betrieb in unzuverlässigen Übertragungsbedingungen mit instabilen Netzen oder stark eingeschränkter Bandbreite. MQe verfügt über ein effizientes Verbindungsprotokoll und automatische Fehlerbehebung bei Kommunikationsverbindungsfehlern.
- Es unterstützt mobile Benutzer, indem es beim Ortswechsel der Geräte den Wechsel der Netzanschlussspunkte ermöglicht. Darüber hinaus ermöglicht MQe auch eine Steuerung des Verhaltens bei schwachen bzw. leeren Batterien und Netzeinschränkungen.
- Es ermöglicht den Einsatz geeignet konfigurierter Firewalls.
- Es minimiert die Verwaltungstasks für den Benutzer. Dadurch ist MQe eine geeignete Basis für die Erstellung von Anwendungen in der Art von Dienstprogrammen.
- Es lässt sich leicht anpassen und erweitern, denn es werden von Anwendungen übergebene *Regeln* verwendet.

MQe unterstützt nicht alle Funktionen von MQ. Einige der wichtigsten Unterschiede (neben umgebungs-, betriebssystem- und übertragungsspezifischen Überlegungen) sind nachfolgend aufgeführt:

- Keine Clusterunterstützung
- Keine Unterstützung von Verteilerlisten
- Keine gruppierten oder segmentierten Nachrichten
- Keine Möglichkeiten zum Lastausgleich bzw. zum Einsatz eines Ersatzsystems (Warm Standby)
- Keine Referenznachrichten
- Keine Berichtsoptionen
- Keine Unterstützung für gemeinsame Warteschlangen
- Keine Auslösefunktion
- Keine UOW-Unterstützung (UOW = Unit of Work; Arbeitseinheit), keine XA-Koordinierung

- Unterschiede bei der Skalierbarkeit und den Leistungsdaten

Allerdings können viele Anwendungstasks in MQe mit alternativen Mitteln bewältigt werden, die MQe-Funktionen verwenden, oder durch die Ausnutzung der Möglichkeit zur Unterklassenbildung, die Ersetzung vorgegebener Klassen oder die Ausnutzung der Regeln, Schnittstellen und sonstigen Anpassungsmöglichkeiten, die in das Produkt eingebaut sind.

Mögliche Einsatzgebiete von MQe

MQe unterstützt mobile Umgebungen und instabile Kommunikationsnetze. Die Systemressourcen werden nur geringfügig belastet, da MQe insbesondere für die Verwendung auf Kleingeräten bestimmt ist. Die Funktionen und Schnittstellen sind genau auf diese Einheiten zugeschnitten, und aus diesem Grund entspricht deren Umfang auch nicht dem sonst bei MQ-Produkten üblichen Umfang. Dafür bietet dieses Produkt jedoch eine Reihe von Sonderfunktionen für diesen bestimmten Benutzerkreis; dazu gehören umfassende Sicherheitsvorrichtungen, Nachrichten, synchrone und asynchrone Nachrichtenübertragung, Zugriff auf ferne Warteschlangen sowie die Nachrichtenübertragung im Push- und Pull-Modus.

Szenarios und Anwendungen

Es sind verschiedene Arten von MQe-Anwendungen möglich, wobei es sich in vielen Fällen um für bestimmte Benutzergruppen entwickelte, speziell angepasste Anwendungen handeln dürfte. Die folgende Liste nennt einige Beispiele:

Handelsanwendungen

- Übertragung einzelner Kassentransaktionen an Hostsysteme wie beispielsweise Nachrichtenbroker

Verbraucheranwendungen

- Lebensmitteleinkäufe von zu Hause über einen PDA
- Ermittlung von Kundenwünschen bei Fluglinien
- Finanzielle Transaktionen über mobile Telefone

Steuerungsanwendungen

- Erfassung und Integration der Daten von Sensoren auf Ölleitungen über Satellit
- Fernsteuerung (z. B. von Ventilen) mit entsprechenden Sicherheitseinrichtungen, die die Berechtigung des Bedieners überprüfen

Mobile Mitarbeiter

- Mitarbeiter im Außendienst (z. B. Versicherungsvertreter)
- Schnelle Übermittlung von Empfangsbestätigungen für Paketzustelldienste
- Kommunikation zwischen Küchenpersonal und Restaurantbedienung
- Punkteählung bei Golfturnieren
- Nachrichtenübertragung über gesicherte mobile Systeme bei der Polizei
- Fachspezifische Informationen für Angestellte bei Versorgungseinrichtungen in Situationen, in denen es zu häufigen Verbindungsabbrüchen kommt
- Zählerablesungen in Privathaushalten

Persönliche Produktivität

- Post-/Kalenderreplikation
- Datenbankreplikation
- Herunterladen von Daten auf Laptops

Wie MQe funktioniert

Die grundlegenden Komponenten des MQe-Programmiermodells sind Nachrichten, Warteschlangen und WS-Manager.

- MQe-Nachrichten enthalten anwendungsdefinierten Inhalt. Die Nachrichten werden in einer Warteschlange gespeichert und können innerhalb eines MQe-Netztes ausgetauscht werden. Dabei werden sie unter Angabe des Ziel-WS-Managers und des Warteschlangennamens an die entsprechende Zielwarteschlange übertragen.
- Anwendungen reihen Nachrichten über *Put*-Vorgänge in Warteschlangen ein und rufen sie über *Get*-Vorgänge wieder ab.
- Bei Warteschlangen kann es sich entweder um *lokale* oder *ferne* Warteschlangen handeln, die von WS-Managern verwaltet werden.
- Die *Registry* speichert Konfigurationsdaten.

Weitere Informationen finden Sie in den übrigen Themen dieses Abschnitts.

Nachrichten

Eine Nachricht ist ein Bündel von Daten, das von einer Anwendung an eine andere geschickt wird. MQe-Nachrichten unterscheiden sich von denen, die durch den Nachrichtenaustausch in MQ unterstützt werden:

- In MQ handelt es sich bei Nachrichten um Bytefeldgruppen (Byte Arrays), die sich aus einem Nachrichtenheader und einem Nachrichtenhauptteil zusammensetzen. Der Nachrichtenheader wird von MQ erstellt und enthält wesentliche Informationen, wie beispielsweise die ID der Warteschlange für Antwortnachrichten, die ID von deren Warteschlangenmanager, die ID der Nachricht und die Korrelations-ID. Der Nachrichtenhauptteil enthält Daten, die nur für die jeweilige Anwendung nützlich sind.
- Bei den Nachrichten in MQe gibt es keine Unterscheidung von Header und Nachrichtenhauptteil. Diese Nachrichten sind vom Typ *MqeFields*, d. h., sie setzen sich aus einem Element vom Typ *name*, einem Element vom Typ *data type* und den eigentlichen Daten zusammen. Die Nachrichtennamen sind ASCII-Zeichenfolgen unbegrenzter Länge mit Ausnahme folgender Zeichen:

{ } [] # () : ; , ' " =

In Tabelle 1 sind die verschiedenen Datentypen beschrieben:

Tabelle 1. Datentypen

Art	Beschreibung
ASCII	Zeichenfolge oder dynamisches Array invarianter ASCII-Zeichenfolgen, ohne die Zeichen { } [] # () : ; , ' " =
Boolescher Wert	Die Werte 'True' (wahr) und 'False' (falsch)
Byte	Fester oder dynamischer Bytewertebereich
Gleitkommazahl mit doppelter Genauigkeit	Wert oder ein fester bzw. dynamischer Bereich von Gleitkommazahlen mit doppelter Genauigkeit
Felder	Objekt oder ein dynamischer Bereich mit Feldobjekten (daher wird die Verschachtelung von Feldobjekten unterstützt)
Gleitkommazahl	Werte oder fester bzw. dynamischer Bereich von Gleitkommazahlen
Integer	4-Byte-Wert oder fester bzw. dynamischer Integerbereich.
Langes Integer	8-Byte-Wert oder fester bzw. dynamischer Bereich langer Integer
Kurzes Integer	2-Byte-Wert oder fester bzw. dynamischer Bereich kurzer Integer
Unicode	Unicode-Zeichenfolge oder dynamischer Bereich von Unicode-Zeichenfolgen

Darüber hinaus beinhalten Nachrichten einen von MQe generierten UID (Unique Identifier, eindeutige Kennung). Dieser UID identifiziert eindeutig jedes einzelne Nachrichtenobjekt im gesamten MQe-Netz und wird aus folgenden Angaben konstruiert:

Quellen-WS-Manager

Dies ist der Name des WS-Managers, von dem die Nachrichtenübertragung ausgeht; der Name muss eindeutig sein. Der Name wird bei Empfang der Nachricht vom WS-Manager hinzugefügt. Da es sich um den Datentyp ASCII handelt, ist jedes Zeichen ein Byte lang.

Erstellungszeit

Dies ist die *Zeitmarke* der Nachricht. Somit ist dies in Java die Zeit auf dem lokalen System, zu der die Nachricht erstellt wurde, während es in C die Zeit auf dem lokalen System ist, zu der ein Feldelement einer Warteschlange hinzugefügt wurde. Aus dem Feldelement wird in diesem Moment eine Nachricht.

Eine Nachricht, die für einen anderen MQe-WS-Manager bestimmt ist, muss keine weiteren Informationen enthalten, obwohl wahrscheinlich weitere Eigenschaften vorhanden sind. Die zusätzlichen Eigenschaften können:

- den aktuellen Status angeben
- an eine bestimmte Nachrichtenunterklasse gebunden sein
- die Anpassung einer Nachricht ermöglichen

Anmerkung: In der C-Codebasis wird ein Feldelement erst zu einer Nachricht, wenn es in eine Warteschlange gestellt wird.

MQe fügt einer Nachricht eigenschaftsbezogene Informationen hinzu bzw. entfernt diese anschließend aus einem Nachrichtenobjekt, um Nachrichtenaustausch- und Warteschlangenoperationen auszuführen. Wenn zwischen zwei WS-Managern eine Nachricht ausgetauscht wird, können Sie Informationen zum erneuten Senden angeben, damit die Daten erneut übertragen werden.

Nachrichten können darüber hinaus *Attribute* haben. Attribute sind von grundlegender Bedeutung für das MQe-Sicherheitskonzept und ermöglichen den selektiven Zugriff auf den Inhalt sowie den Schutz des Inhalts. Sie verfügen über die folgenden Eigenschaften:

Tabelle 2. Attributobjekt - Eigenschaften

Eigenschaft	Beschreibung
Authentifizierung	Ermöglicht die Zugriffssteuerung
Verschlüsselung	Schützt den Objekteinhalt bei Ausgabe in einen Bytebereich (und erlaubt die Wiederherstellung dieses Inhalts)
Komprimierung	Reduziert den Speicherbedarf (für Übertragung und Speicherung)
Regel (nicht zutreffend für die C-Codebasis)	Ermöglicht die Steuerung zulässiger Vorgänge

Weitere Informationen zu den Eigenschaften in Tabelle 2 finden Sie im Abschnitt „Sicherheit“ auf Seite 21.

Warteschlangen

Warteschlangen dienen zur Aufnahme von Nachrichten. Anwendungen greifen auf Warteschlangen nicht direkt zu, sondern über den Warteschlangenmanager.

Warteschlangen werden über ihren Namen identifiziert, bei dem es sich um eine ASCII-Zeichenfolge mit unbegrenzter Länge ohne die folgenden Zeichen handeln kann:

{ } [] # () : ; , ' " =

Warteschlangennamen müssen jedoch innerhalb eines WS-Managers eindeutig sein. Zur Interoperabilität mit MQ wird empfohlen, Namensbeschränkungen von MQ einzuhalten, insbesondere die maximale Namenslänge von 48 Zeichen. Die Länge des Namens kann auch durch das von Ihnen verwendete Dateisystem eingeschränkt werden. MQe unterstützt eine Reihe verschiedener Warteschlangenarten:

Lokale Warteschlangen

Lokale Warteschlangen werden von Anwendungen zur sicheren Nachrichtenspeicherung verwendet (wobei Hardwarefehler und der Ausfall von Einheiten nicht ausgeschlossen werden können). Lokale Warteschlangen gehören zu einem bestimmten WS-Manager. Dabei kann es sich entweder um einen Standalone-WS-Manager oder einen in ein Netzwerk eingebundenen WS-Manager handeln.

Ferne Warteschlangen

Bei fernen Warteschlangen handelt es sich um lokale Verweise auf Warteschlangen, die sich in einem fernen WS-Manager im MQe-Netz befinden. Der lokale Verweis hat denselben Namen wie die Zielwarteschlange, allerdings gibt die Definition der fernen Warteschlange den WS-Manager an, dem die tatsächliche Warteschlange zugeordnet ist. Ferne Warteschlangen verfügen darüber hinaus über zugriffsspezifische Eigenschaften sowie sicherheitsspezifische Merkmale und Übertragungsoptionen. Ihr Zugriffsmodus kann synchron oder asynchron sein. Synchrone ferne Warteschlangen verfügen über keinen Speicher; die Nachricht wird unmittelbar an den fernen Warteschlangenmanager weitergeleitet. Eine asynchrone Warteschlange speichert die Nachricht im lokalen Speicher; sie wird dann von einem Hintergrundthread an den fernen Warteschlangenmanager gesendet.

Warteschlangen zum Speichern und Weiterleiten

Eine Warteschlange zum Speichern und Weiterleiten übernimmt die Zwischenspeicherung von Nachrichten für einen oder mehrere WS-Manager, bis dieser oder diese die Nachrichten empfangen können. Diese Art von Warteschlange ist in der C-Codebasis nicht vorhanden. Warteschlangen zum Speichern und Weiterleiten werden hauptsächlich zu den folgenden zwei Zwecken verwendet:

1. Sie ermöglichen zum einen, dass Nachrichten im Netz zwischengespeichert und so etappenweise an ihr Ziel weitergeleitet werden können.
2. Zum anderen ermöglichen sie das Speichern von Nachrichten, bis diese von einer Ausgangs-server-Warteschlange erfasst werden.

Diese Warteschlange wird in der Regel (jedoch nicht notwendigerweise) auf einem Server oder Gateway definiert. Warteschlangen zum Speichern und Weiterleiten können sowohl mehreren Ziel-WS-Managern gleichzeitig als auch jeweils nur einem Ziel-WS-Manager zugeordnet werden.

Ausgangsserver-Warteschlangen

Ferne Warteschlangen und Warteschlangen zum Speichern und Weiterleiten senden Nachrichten im Push-Modus über das Netz, wobei die Übertragung von der sendenden Warteschlange eingeleitet wird. Ausgangsserver-Warteschlangen rufen dagegen Nachrichten im Pull-Modus von einer fernen Warteschlange ab. Nachrichten werden deshalb nie an Ausgangsserver-Warteschlangen gerichtet.

Die Definition einer Ausgangsserver-Warteschlange gibt eine Warteschlange zum Speichern und Weiterleiten auf einem fernen WS-Manager an. Die Ausgangsserver-Warteschlange ruft anschließend die Nachrichten, die an ihren lokalen WS-Manager gerichtet sind, im Pull-Modus aus der Warteschlange zum Speichern und Weiterleiten ab. Ein einzelner WS-Manager kann mehrere Definitionen von Ausgangsserver-Warteschlangen enthalten, die jeweils einer anderen fernen Warteschlange zum Speichern und Weiterleiten zugeordnet sind.

Verwaltungswarteschlangen

Bei der Verwaltungswarteschlange handelt es sich um eine lokale Warteschlange, die Verwaltungsnachrichten empfängt. Eine Verwaltungsnachricht enthält Anweisungen, die intern von der

Anwendung ausgeführt werden und sich auf eine bestimmte Komponente von MQe beziehen. Für jede Verwaltungsaktion kann optional eine Verwaltungsnachricht als Antwort angefordert werden, die zurück an die Ursprungsanwendung gesendet wird. Diese Antwortnachrichten informieren Sie über Erfolg oder Fehlschlag der jeweiligen Verwaltungsaktion. Auf diese Weise kann durch die Verwendung von Verwaltungswarteschlangen von einer Komponente auf einem WS-Manager die Konfiguration eines zweiten WS-Managers entweder synchron oder asynchron gesteuert werden. Verwaltungsnachrichten werden in derselben Reihenfolge verarbeitet, wie sie in der Verwaltungswarteschlange eingetroffen sind. Weitere Informationen hierzu finden Sie im Abschnitt „Verwaltung“ auf Seite 15.

MQ-Brückenwarteschlangen

Eine Brückenwarteschlange ist eine Sonderform der fernen Warteschlange, womit eine Warteschlange auf einem fernen MQ-WS-Manager gemeint ist. Brückenwarteschlangen reihen Nachrichten in die MQ-Warteschlange ein, auf die sie verweisen, oder rufen sie von dort ab. Nur in Java wird dabei ein Transformator verwendet, mit dem bei jedem Austausch von Nachrichten zwischen den MQe- und den MQ-Systemen die erforderliche Neuformatierung von Daten oder Nachrichten ausgeführt wird. Brückenwarteschlangen können nur auf einem Gateway-WS-Manager erstellt werden.

MQe speichert Daten geschützt in Warteschlangen und sorgt dafür, dass Nachrichten physisch auf den Datenträger geschrieben und nicht nur vom Betriebssystem gespeichert werden. Allerdings führt MQe kein unabhängiges Protokoll über Änderungen an Nachrichten und Warteschlangen. Für eine Wiederherstellung nach einem Datenträgerfehler müssen daher Hardwarelösungen wie z. B. RAID-Datenträgersysteme eingesetzt werden. Als Alternative dazu kann die Warteschlange einem wiederherstellbaren Speicherbereich wie z. B. den Datenbanksystemen zugeordnet werden. Die MQe-Clients sind auf Kleingerätesystemen angesiedelt, wohingegen Server-Warteschlangenmanager oft rund um die Uhr verfügbar sein müssen, so dass die Übernahme durch einen Ersatz bei Ausfall vorgesehen sein muss.

MQe umfasst vier allgemein verwendete Systemwarteschlangen:

Verwaltungswarteschlange

Empfängt Verwaltungsnachrichten

Warteschlange für nicht zustellbare Nachrichten

Speichert Nachrichten, die ansonsten nicht zugestellt werden können

Warteschlange für Verwaltungsantwortnachrichten

Empfängt Antworten auf Verwaltungsnachrichten

SYSTEM.DEFAULT.LOCAL.QUEUE

Hat denselben Namen wie die obligatorische Systemwarteschlange auf MQ-Servern

WS-Manager

Über den MQe-WS-Manager kann MQe verschiedene Netzkonfigurationen unterstützen. Dies umfasst Folgendes:

- Einen zentralen Zugriffspunkt für ein Netzwerk zum Nachrichtenaustausch und zur Warteschlangeneinreihung für MQe-Anwendungen
- Optionale clientseitige Warteschlangeneinreihung
- Verbindungssteuerung
- Optionale Verwaltungsfunktionen
- Einmalige gesicherte Zustellung von Nachrichten
- Automatische Wiederherstellung nach Fehlern und Ausfällen
- Benutzerdefiniertes regelbasiertes Verhalten

In MQE kann immer nur ein WS-Manager auf einer einzigen JVM (Java Virtual Machine) bzw. in einem einzigen nativen Anwendungsprozess aktiv sein. Wenn mehrere WS-Manager auf einer Maschine ausgeführt werden sollen, müssen mehrere JVM-Exemplare bzw. mehrere native Anwendungsprozesse vorhanden sein.

WS-Manager werden identifiziert durch einen systemweit eindeutigen Namen und eine ASCII-Zeichenfolge unbegrenzter Länge ausschließlich der folgenden Zeichen:

{ } [] # () : ; , ' " =

Die Einhaltung dieser Einschränkung wird durch MQE bzw. MQ nicht überprüft; allerdings können doppelte WS-Managernamen dazu führen, dass Nachrichten an den falschen WS-Manager übertragen werden. Um die Zusammenarbeit mit anderen Produkten der Produktfamilie zu erleichtern, sollten die Namen maximal 48 Zeichen lang sein. Die Länge des Namens kann außerdem durch das von Ihnen verwendete Dateisystem eingeschränkt werden.

WS-Manager können mit oder ohne lokale Warteschlangensteuerung konfiguriert werden. Alle WS-Manager unterstützen die Operationen des synchronen Nachrichtenaustauschs. WS-Manager mit lokaler Warteschlangensteuerung unterstützen auch die asynchrone Nachrichtenzustellung. Zwischen der asynchronen und der synchronen Nachrichtenzustellung bestehen erhebliche Unterschiede in Charakteristik und Auswirkungen:

Synchrone Nachrichtenzustellung

Bei der synchronen Nachrichtenzustellung übergibt die Anwendung die Nachricht an MQE zur Übertragung an die ferne Warteschlange. MQE setzt sich gleichzeitig mit der Zielwarteschlange in Verbindung und reiht die Nachricht ein. Im Anschluss an die Zustellung kehrt MQE sofort zur Anwendung zurück. Kann die Nachricht nicht zugestellt werden, wird umgehend eine entsprechende Meldung an die sendende Anwendung geschickt. Im synchronen Fall übernimmt MQE keine Verantwortung für die Zustellung der Nachricht (nicht garantierte Zustellung).

Asynchrone Nachrichtenzustellung

Bei der asynchronen Nachrichtenzustellung übergibt die Anwendung die Nachricht an MQE zur Übertragung an eine ferne Warteschlange. MQE kehrt sofort zur Anwendung zurück. Kann die Nachricht sofort zugestellt oder in einen zwischengeschalteten Puffer eingereiht werden, wird sie gesendet. Andernfalls wird sie lokal zwischengespeichert. Die asynchrone Zustellung sorgt für die einmalige garantierte Nachrichtenzustellung, da die Nachricht an MQE übergeben wird und daher MQE für die Zustellung zuständig ist (garantierte Nachrichtenzustellung).

WS-Managerkonfiguration: Ein MQE-Warteschlangenmanager benötigt eine Anwendung, die die erforderliche Umgebung erstellt, bevor der Warteschlangenmanager geladen wird. Das bedeutet, dass die Anwendung, die den Warteschlangenmanager startet, Zugriff auf bestimmte Informationen haben muss, bevor sie den Warteschlangenmanager laden kann. Wenn Sie z. B. die Paketgröße für den Kommunikationsadapter in Java setzen, muss die erforderliche Java-Eigenschaft vor dem Laden des Warteschlangenmanagers gesetzt werden. Die Anwendung benötigt folgende zwei Angaben: wo sich die MQE Registry befindet, und wo sich der Warteschlangenspeicher befindet. Die Registry (nicht zu verwechseln mit der Windows Registry) ist der Ort, an dem die Definition aller zum Warteschlangenmanager gehörenden Objekte liegt, z. B. Instanzwarteschlangen und Verbindungsdefinitionen. Dadurch kann ein Warteschlangenmanager die richtigen Objekte erzeugen, wenn er geladen wird. Der Warteschlangenspeicher ist der Ort, an dem sich die Warteschlangen befinden; durch ihn kann ein Warteschlangenmanager Nachrichten in lokalen Warteschlangen aufbewahren, die zwischen dem Stoppen und erneuten Starten eines Warteschlangenmanagers bestehen bleiben.

Die MQE-Umgebung können Sie mit Hilfe der Anwendungsprogrammierschnittstelle (API), der im Lieferumfang von MQE enthaltenen Dienstprogramme oder von Verwaltungstools wie z. B. MQE_Explorer konfigurieren. Diese Tools können die Umgebungsparameter in einer Initialisierungsdatei aufzeichnen; dies ist jedoch optional.

Sie können einen Warteschlangenmanager so konfigurieren, dass er als MQ-Brücke fungiert. So ein Warteschlangenmanager wird als Gateway bezeichnet und ermöglicht in Java den Austausch von Nachrichten mit Hostprodukten und verteilten Produkten von MQ. In der C-Codebasis wird nur ein Geräte-WS-Manager verwendet.

Warteschlangenmanager-Operationen: WS-Manager unterstützen Vorgänge im Zusammenhang mit der Nachrichtenübertragung und können für die Warteschlangenverwaltung eingesetzt werden. Anwendungen greifen über die Services des WS-Managers auf Nachrichten zu; dabei verwenden sie u. a. folgende Methoden:

Get (Abrufen)

Bei diesem Vorgang werden Nachrichten aus einer Warteschlange entfernt.

Put (Einreihen)

Bei diesem Vorgang werden Nachrichten in eine Warteschlange eingereiht.

Delete (Löschen)

Durch Angabe des UIDs können die jeweiligen Nachrichten aus einer Warteschlange gelöscht werden, ohne die Get-Methode zu verwenden.

Browse (Durchsuchen)

Sie können Warteschlangen mit Hilfe eines *Filters* nach bestimmten Nachrichten durchsuchen (siehe unten). Bei diesem Vorgang werden alle Nachrichten abgerufen, die mit den Filterkriterien übereinstimmen, sie sind jedoch weiterhin in der Warteschlange vorhanden. MQe unterstützt außerdem die Funktion 'Durchsuchen mit Sperre' (Browsing under lock). Mit dieser Funktion können die mit den Suchkriterien übereinstimmenden Nachrichten gesperrt werden.

Wait (Warten)

In Java können Anwendungen eine bestimmte Zeit auf die Einreihung einer Nachricht in eine Warteschlange *warten*. Dies gilt jedoch nicht für die C-Codebasis.

Listen (Überwachen)

In Java können Anwendungen auch das Auftreten von MQe-Nachrichtenereignissen überprüfen; auch hier ist die Angabe eines Filters möglich. Dazu muss der Warteschlange jedoch eine Empfangsfunktion hinzugefügt werden. Empfangsfunktionen werden beim Eingang von Nachrichten in einer Warteschlange entsprechend informiert. Dies gilt jedoch nicht für die C-Codebasis.

Für eine Reihe dieser Vorgänge kann als einer ihrer Parameter ein **Filter** angegeben werden. Ein Filter überprüft ein Element auf Übereinstimmung, und alle Teile der Nachricht können unabhängig von den anderen Teilen abgerufen werden. Die meisten Methodenaufrufe können darüber hinaus ein Attribut enthalten, das für die Ver- bzw. Entschlüsselung einer Nachricht verwendet wird.

Verwaltung

Die Verwaltung wird über die MQe-Schnittstelle durch Generieren und Empfangen von Verwaltungsnachrichten abgewickelt. Während Anwendungen die nachrichtenbezogenen Funktionen übernehmen, stellt die Verwaltung Funktionen zum Konfigurieren und Verwalten von MQe-Ressourcen wie z. B. Warteschlangen und Verbindungen bereit.

Anforderungen werden an die Verwaltungwarteschlange auf dem Ziel-WS-Manager gesendet; bei Bedarf können Antworten zurückgegeben werden. Jedes lokale oder ferne MQe-Anwendungsprogramm kann *Verwaltungsnachrichten* direkt oder indirekt über Hilfemethoden erstellen und verarbeiten.

Die C-Codebasis bietet einen Administrator, der Ihnen die Durchführung einiger Verwaltungsmaßnahmen ermöglicht. Diese Aktionen werden nur für Ressourcen ausgeführt, die vom lokalen WS-Manager verwaltet werden.

Die Verwaltungwarteschlange selbst kann die Verwaltung einzelner Ressourcen nicht ausführen. Die relevanten Daten sind in der jeweiligen Ressource und der zugehörigen Verwaltungsnachricht enthalten.

Verwaltungsnachrichten: Nach dem Erstellen der WS-Manager werden sie konfiguriert, indem der Verwaltungswarteschlange auf dem Ziel-WS-Manager die entsprechenden Verwaltungsnachrichten gesendet werden. Ein WS-Manager, der keine Verwaltungswarteschlange hat, kann nicht verwaltet werden. Die Verwendung von Verwaltungsnachrichten hat jedoch den Vorteil, dass die lokale und ferne Verwaltung auf dieselbe Art und Weise ausgeführt werden kann.

Eine Verwaltungsnachricht wird erstellt und an die Verwaltungswarteschlange auf dem WS-Manager gesendet, der verwaltet werden soll. Für die Zugriffssteuerung können warteschlangenbasierte Sicherheitsattribute verwendet werden. Eine Verwaltungsnachricht enthält Informationen zur Anforderung, die Angabe, ob eine Antwort erforderlich ist oder nicht, sowie die Adresse des Ziel-WS-Managers und der Warteschlange. In MQe können daher folgende Arten von Verwaltungsnachrichten verwendet werden:

- Befehle, mit denen eine Verwaltungsaktion angegeben wird, für die keine Antwort erforderlich ist.
- Anforderungen, für die eine Antwort erforderlich ist.
- Antwortnachrichten, die auf der Basis einer Kopie der Originalnachricht erstellt wurden.

Der Absender kann zusätzliche Felder hinzufügen, die der Empfänger verwenden kann. Die Verwaltungswarteschlange selbst reagiert auf die Nachricht. Mit Verwaltungsnachrichten können Informationen zu Objekten angefordert sowie Objekte erstellt, gelöscht oder aktualisiert werden. Für eine Untergruppe der Objekte können darüber hinaus zusätzliche Funktionen wie z. B. das Stoppen und Starten von Objekten mit Hilfe der Nachrichten ausgeführt werden.

Verwaltungsnachrichten können außerdem auch indirekt über MQe_Explorer erstellt werden. Dabei handelt es sich um ein Verwaltungstool mit einer grafischen Benutzeroberfläche für die Systemverwaltung. MQe_Explorer gehört nicht zum Lieferumfang von MQe, steht aber zum kostenlosen Download als SupportPac zur Verfügung.

Selektive Verwaltung: Über die Authentifizierungsfunktion der Verwaltungswarteschlange wird der Zugriff auf die Verwaltung gesteuert. Die bereitgestellte Authentifizierungsfunktion ordnet alle lokalen Anwendungen demselben lokalen Benutzer zu und ermöglicht bzw. sperrt daher den Zugriff auf die Verwaltung all dieser Anwendungen.

Wird die Authentifizierungsfunktion jedoch für die Verbindung gestartet, noch bevor die ersten Verwaltungsnachrichten gesendet werden, gilt die Zugriffssteuerung für die fernen Verwaltungsanwendungen. Dabei werden die verschiedenen fernen Anwendungen getrennt erfasst, so dass der Verwaltungszugriff für jede ferne Anwendung einzeln ermöglicht bzw. gesperrt werden kann. In jedem Fall wird die Verwaltung jeweils vollständig ermöglicht bzw. vollständig gesperrt.

Eine Authentifizierungsfunktion verfügt über eine Übersicht über die den verschiedenen Benutzer-IDs zugeordneten Berechtigungen; daher können Verwaltungsnachrichten anhand dieser Berechtigungen verarbeitet werden. Weitere Informationen zur Authentifizierung finden Sie im Abschnitt „Sicherheit“ auf Seite 21. Durch Verwendung von Regeln, die den Warteschlangen zugeordnet werden, können auf gleiche Weise auch Aktionen ermöglicht bzw. gesperrt werden. Weitere Informationen zu Regeln finden Sie im Abschnitt „Regeln für benutzerdefinierte Konfiguration“ auf Seite 23.

Überwachung und damit zusammenhängende Vorgänge: Zur Verwaltung gehört mehr als das Erstellen und Ändern von Elementen. Sie umfasst auch die Überwachung eines Systems, Meldungen an den Bediener, wenn eine Warteschlange voll ist, oder auch die Fehlerbehebung wie beispielsweise die Ausführung entsprechender Maßnahmen bei Eingang einer Nachricht, die zu groß für die Zielwarteschlange ist. Diese Aufgaben werden in MQe über Regeln gesteuert, sobald größere Änderungen am Status eines Elements oder bestimmte Fehler auftreten. MQe stellt Standardregeln zur Implementierung bereit, die jedoch bei Bedarf angepasst werden können. Weitere Informationen dazu finden Sie im Abschnitt „Regeln für benutzerdefinierte Konfiguration“ auf Seite 23.

Verbindungen

Eine Verbindung stellt einem WS-Manager die Informationen bereit, die für die Kommunikation mit einem fernen WS-Manager erforderlich sind. Die WS-Manager verwenden die Verbindung dann zum Informationsaustausch. Verbindungsdefinitionen werden lokal auf dem jeweiligen WS-Manager gespeichert.

Anmerkung: Die C-Codebasis ist nur ein Geräte-WS-Manager.

Hier einige der wichtigsten Merkmale von Verbindungen:

Unterstützung sowohl von *synchronem* als auch von *asynchronem* Nachrichtenaustausch

Die synchrone Nachrichtenübertragung ermöglicht einen direkten Übertragungsservice von der Quellenanwendung zur Zielwarteschlange, ohne dass die Nachrichten im Quellen-WS-Manager eingereicht werden müssen. Die asynchrone Nachrichtenübertragung bietet einen Übertragungsservice vom Quellen-WS-Manager zur Zielwarteschlange, wobei die Nachrichten im Quellen-WS-Manager eingereicht werden können.

Bereitstellung von *End-to-End-Service*

Die Verbindungen führen vom Quellen- zum Ziel-WS-Manager, wobei zwischengeschaltete WS-Manager möglich sind. Das in der Verbindung verwendete Übertragungsprotokoll kann sich auf dem Weg durch diese zwischengeschalteten WS-Manager ändern. Mehrere Verbindungen können zu einer Verbindung für die Endpunkt-zu-Endpunkt-Übertragung zusammengeschlossen werden.

Unterstützung von *Komprimierung, Verschlüsselung und Authentifizierung*

Verbindungen bieten diese Sicherheitsmerkmale, um die Daten bei der Übertragung zu schützen.

Unterstützung von *Client/Server-Betrieb*

Client/Server-Verbindungen funktionieren nach dem Anforderungs-/Antwortprinzip. Der Client sendet eine Anforderung an den Server, und der Server antwortet auf die Anforderung. Dies bedeutet jedoch keine Einschränkung für den Nachrichtenfluss. Es können Nachrichten vom Client an den Server fließen und vom Server an den Client.

Die folgenden Diagramme enthalten Beispiele für typische MQE-Konfigurationen. Aus Gründen der Übersichtlichkeit sind nur die definierten Direktverbindungen zu sehen. Es besteht jedoch auch die Möglichkeit, indirekte Verbindungen über Direktverbindungen zu definieren. In den Diagrammen werden Client/Server-Verbindungen durch einen Pfeil dargestellt, der auf den Server gerichtet ist. Clients können die Client/Server-Verbindungen sowohl für die Nachrichtenübertragung an den Server als auch für den Nachrichtenempfang vom Server im Pull-Modus verwenden. Verbindungslinien ohne Pfeil stehen für MQ-Clientkanäle, die eine Kommunikation zwischen MQE und MQ ermöglichen.



Abbildung 1. Standalone-MQe-Warteschlangenmanager

In Abb. 1 ist ein Standalone-WS-Manager zu sehen, der eine oder mehrere Anwendungen unterstützt, die für den Datenaustausch Warteschlangen verwenden.

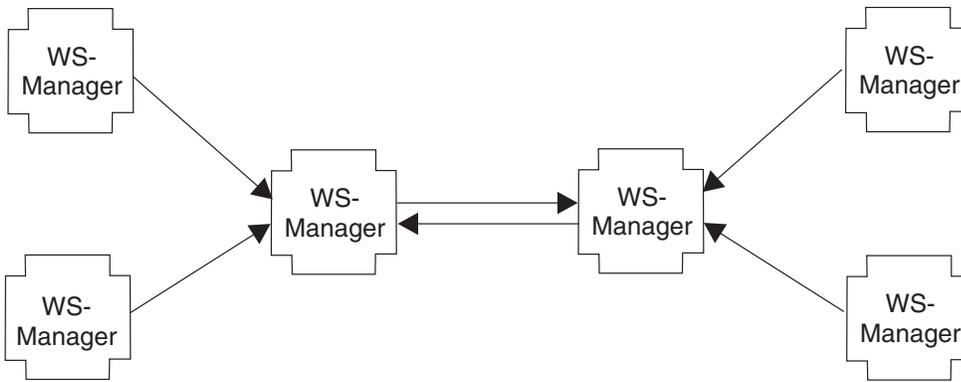


Abbildung 2. Konfiguration kleiner Netzwerke

In Abb. 2 ist die Konfiguration eines kleinen Netzwerks zu sehen; die zentralen Server-WS-Manager tauschen über zwei direkte Client/Server-Verbindungen Daten aus. Die Client-WS-Manager verwenden jeweils eine direkte Client/Server-Verbindung für den Datenaustausch mit den Server-WS-Managern.

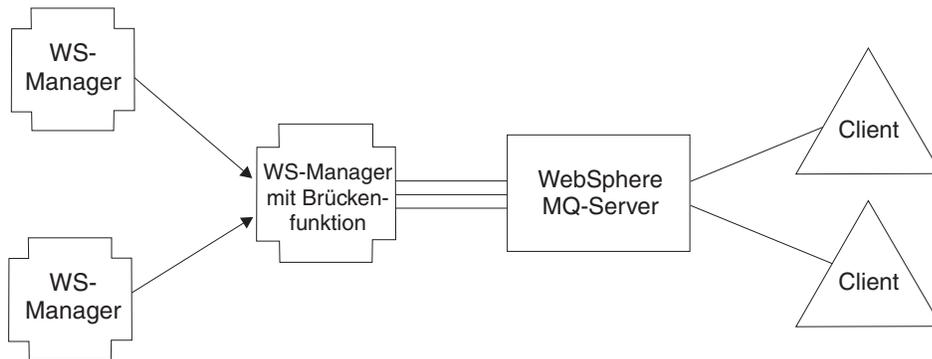


Abbildung 3. Integriertes MQ-Familien-Netzwerk

In Abb. 3 ist eine MQE-Konfiguration zu sehen, bei der bei einem der WS-Manager die optionale Brückenfunktion eingestellt ist; der Pool der Clientkanäle ist einem einzelnen MQ-Zielserver in einer Hostumgebung oder in einer verteilten Umgebung zugeordnet.

Da Verbindungen in der Regel den Zugriff auf einen fernen WS-Manager definieren, werden sie bisweilen als "Definitionen ferner WS-Manager" bezeichnet.

Es können auch indirekte Verbindungen definiert werden. In diesem Fall leitet MQE die Verbindung über andere WS-Manager (die miteinander verkettet werden können) weiter; das Übertragungsprotokoll kann sich auf dem Übertragungsweg ändern. Indirekte Verbindungen sind besonders dadurch von Nutzen, dass sie es ermöglichen, für Geräte einen zentralen Zugang zu einem Netzwerk einzurichten.

Wie für die meisten MQ-Komponenten können Sie auch für Verbindungen Aliasnamen definieren. Verwenden Sie eine lokale Verbindung (als Verbindung mit einem Namen definiert, der dem des lokalen WS-Managers entspricht), um für den lokalen WS-Manager selbst Aliasnamen zu definieren.

Verbindungen unterstützen die bidirektionale Übertragung und werden bei Bedarf vom WS-Manager eingerichtet. Die asynchrone und die synchrone Nachrichtenübertragung erfolgen über dieselben Verbindungen; das dazu verwendete Protokoll ist MQE-spezifisch.

Über die Verbindungsdefinitionen werden die Leitungen und Protokolle festgelegt, die für eine bestimmte Verbindung verwendet werden sollen. An jedem zwischengeschalteten Knoten werden die übermittelten Nachrichten an den an diesem Punkt vorhandenen WS-Manager übergeben. Die Handhabung der Nachrichten durch den WS-Manager hängt von dessen Ressourcen ab. Eine Nachricht wird in eine Warteschlange eingereiht, bei der es sich um eine lokale Warteschlange, eine ferne Warteschlange oder eine Warteschlange zum Speichern und Weiterleiten handeln kann. Bei Nachrichten, die in ferne Warteschlangen eingereiht wurden, hängt die anschließende Weiterleitung von der Art der fernen Warteschlange ab. Ferne Warteschlangen für synchrone Nachrichtenübertragung leiten die Nachrichten unmittelbar weiter. Ferne Warteschlangen für asynchrone Nachrichtenübertragung speichern die Nachrichten, bevor sie diese weiterleiten.

Verbindungen sind für Anwendungsprogrammierer und Administratoren nicht direkt sichtbar und werden vom WS-Manager bei Bedarf erstellt. Verbindungen stellen den Übertragungsweg zwischen WS-Managern dar, und ihre Merkmale werden von MQe je nach den zu übertragenden Informationen geändert. Bei Übertragungsprogrammen handelt es sich um MQe-Komponenten, die unter Verwendung der Verbindungen eine Kommunikation auf Warteschlangenebene ermöglichen. Auch diese Übertragungsprogramme sind für den Anwendungsprogrammierer oder Administrator nicht sichtbar.

Wenn die garantierte Nachrichtenübertragung gefordert wird, werden die Nachrichten nur einmal von MQe an die Anwendung zugestellt. Dies wird dadurch erreicht, dass die Kopie einer Nachricht auf der Senderseite erst gelöscht wird, nachdem die Nachricht erfolgreich von einem WS-Manager zum anderen übertragen und bestätigt wurde. Wird im Fall eines Kommunikationsfehlers keine Empfangsbestätigung empfangen, kann eine Nachricht noch einmal übertragen werden, da einmalige *Zustellung* nicht gleichbedeutend mit einmaliger *Übertragung* ist. Nachrichten werden jedoch nicht mehrfach zugestellt.

Verbindungsarten: MQe unterstützt einen Client/Server-Betrieb. Ein *Client* kann die Kommunikation mit einem Server einleiten. Ein *Server* kann hingegen nur auf die von einem Client gesendeten Anforderungen reagieren. Hierbei handelt es sich um folgende Komponenten:

Empfangsfunktion

Ist für eingehende Verbindungsanforderungen empfangsbereit.

WS-Manager

Unterstützt Anwendungen über die Steuerung der Nachrichtenübertragung und die Steuerung von Warteschlangen.

Tabelle 3. Verbindungsarten

	WS-Manager	Empfangsfunktion
Client	Ja	Nein
Client/Server	Ja	Ja
Server	Ja	Ja
Servlet	Ja	Nein

In Tabelle 3 ist aufgeführt, welche Komponenten für welche Verbindungsart erforderlich sind. Bei der Client/Server-Verbindungsart kann MQe im Client- oder im Servermodus betrieben werden. Bei der Servlet-Option wird MQe als HTTP-Servlet konfiguriert, wobei der HTTP-Server selbst für eingehende Verbindungsanforderungen empfangsbereit sein muss.

MQe-Anwendungen ist nicht direkt bekannt, welche Verbindungsart von den Warteschlangenmanagern verwendet wird. Allerdings kommt es auf die Verbindungsart insofern an, als sie sich darauf auswirkt, welche Ressourcen den beteiligten Seiten zur Verfügung stehen, welche Warteschlangenmanager mit anderen Warteschlangenmanagern Verbindung aufnehmen können, wie viel Speicher MQe verwendet und welche Verbindungen gleichzeitig bestehen können.

Adapter

Adapter werden für die Zuordnung von MQe zu Geräteschnittstellen verwendet. Beispiel:

- Kanäle verwenden Protokolladapter, um HTTP, natives TCP/IP, UDP und andere Protokolle für die Übertragung einzusetzen.
- Warteschlangen nutzen die Feldspeicheradapter als Schnittstelle zu Speichersubsystemen wie beispielsweise dem Hauptspeicher oder dem Dateisystem.

Durch Adapter ist ein Mechanismus verfügbar, mit dem MQe seine Geräteunterstützung erweitern und Versionskontrolle ermöglichen kann.

Anmerkung: Im Unterschied zur Java Codebasis von MQe verwendet die C-Codebasis nur den HTTP-Adapter.

Verwaltung von Einwählverbindungen

Für die Unterstützung der Netzeinwahl für Einheiten ist das Betriebssystem der Einheit verantwortlich.

Versucht MQe auf einer nicht verbundenen Einheit einen Netzzugriff durchzuführen (z. B. um eine Nachricht zu übertragen) und ist der Netzstapel nicht aktiv, wird RAS (Remote Access Services) vom Betriebssystem selbst initialisiert. In der Regel wird dabei eine Anzeige mit einem Wählverbindungsprofil für den Benutzer geöffnet.

Das Betriebssystem übernimmt die Steuerung, bis eine Verbindung hergestellt wurde. Daher muss der Einheitenbenutzer sicherstellen, dass für das Betriebssystem die entsprechenden Wählverbindungsprofile vorhanden sind. Es gibt keine direkte Unterstützung für den Wählnetzbetrieb in MQe.

Trace

Ein unabhängiges Programm ist für die Tracedurchführung zuständig.

Aufrufe zum Erfassen von Situationen mit Informationsnachrichten, Warnnachrichten und Fehlermeldungen in einem Trace sind in MQe eingebettet.

Anwendungen können den Trace auch direkt aufrufen, und sie können, allerdings nur bei Verwendung der Java-Codebasis von MQe, Nachrichten hinzufügen.

Da die Schnittstelle, die von einer Tracesteueroutine implementiert werden muss, veröffentlicht wird, können Lösungen diese Schnittstelle implementieren, um den MQe- und Anwendungs-Trace abzurufen, einzubinden und die Ausgabe an eine Stelle zu übertragen, von der sie abgerufen werden kann. Als Teil des Produktcodes stehen mehrere Tracesteueroutinen zur Verfügung.

Da darüber hinaus der Großteil der MQe-Ausnahmebedingungen zur Verarbeitung an die Anwendung übergeben wird, kann die Ausnahmebehandlungsroutine der Anwendung diese Ausnahmebedingungen ebenfalls an das Traceprogramm weiterleiten.

Ereignisprotokoll

Dieser Abschnitt gilt nicht für die C-Codebasis.

MQe stellt Ereignisprotokollmechanismen und -schnittstellen zur Verfügung, mit denen Statusinformationen protokolliert werden können. Beispielsweise werden Warnnachrichten protokolliert, wenn die Nachrichten in einer asynchronen fernen Warteschlange nicht zugestellt werden können.

Standardmäßig wird das Protokoll in ein `system.out` geschrieben, doch es kann abgefangen und an eine andere Stelle umgeleitet werden.

Im MQe-Ereignisprotokoll werden keine Nachrichtendaten protokolliert, und es kann nicht zur Wiederherstellung von Nachrichten oder Warteschlangen verwendet werden.

Sicherheit

In MQe stehen integrierte Sicherheitseinrichtungen zur Verfügung, die den Schutz von Nachrichten sowohl im lokal gespeicherten Zustand als auch bei der Übertragung ermöglichen.

Die MQe-Sicherheitseinrichtungen können in drei Kategorien eingeteilt werden:

Sicherheit auf lokaler Ebene

Schützt nachrichtenbezogene Daten auf lokaler Ebene.

Sicherheit auf Nachrichtenebene

Schützt Nachrichten zwischen der Ursprungsanwendung und der empfangenden Anwendung von MQe.

Sicherheit auf Warteschlangenebene

Schützt Nachrichten zwischen dem Ursprungs-WS-Manager und der Zielwarteschlange.

Die Sicherheit auf lokaler Ebene und die Sicherheit auf Nachrichtenebene werden von MQe intern genutzt und können auch für MQe-Anwendungen verfügbar gemacht werden. Bei der Sicherheit auf Warteschlangenebene von MQe handelt es sich um einen internen Service.

Die MQe-Sicherheitseinrichtungen aller drei Kategorien schützen die Nachrichtendaten unter Verwendung eines Attributs, z. B. MQeAttribute. Je nach Kategorie wird das Attribut extern oder intern verwendet.

Jedes Attribut kann Folgendes enthalten:

Authentifizierungsfunktion

Stellt zusätzliche Steuerelemente bereit, um den Zugriff nicht berechtigter Benutzer auf lokale Daten zu vermeiden.

Verschlüsselungsfunktion

Steuert den erforderlichen Umfang der Schutzmaßnahmen.

Komprimierungsfunktion

Optimiert die Größe der geschützten Daten.

Schlüssel

Steuert den Zugriff durch Aufforderung zur Eingabe eines Kennworts.

Name der Zieleinheit

Fordert zur Eingabe des Namens der Zielwarteschlange auf.

Diese Elemente werden je nach der Sicherheitskategorie von MQe unterschiedlich verwendet; in allen Fällen wird jedoch der Schutzmechanismus der jeweiligen MQe-Sicherheitseinrichtung aktiviert, wenn das einer Nachricht zugeordnete Attribut aufgerufen wird.

Die Registry: Die Registry ist der wichtigste Speicherort für die auf WS-Manager bezogenen Informationen. Alle WS-Manager verfügen jeweils über mindestens eine Registry. Jeder WS-Manager speichert folgende Daten in dieser Datenbank:

- Konfigurationsdaten des WS-Managers
- Warteschlangendefinitionen
- Definitionen ferner Warteschlangen
- Verbindungsdefinitionen
- Benutzerdaten (einschließlich konfigurationsspezifischer Sicherheitsdaten)

Die Daten der Registry werden mit Hilfe eines Adapters gespeichert, wobei es sich in der Regel um den Adapter 'MQeDiskFields' handelt.

Private Registry und Berechtigungen: Dieser Abschnitt gilt nicht für die C-Codebasis.

Da jede Einheit eigene Berechtigungen für die Authentifizierung benötigt, muss Folgendes bekannt sein:

1. Vorgehensweise zur Registrierung, damit die Berechtigungen abgerufen werden können
2. Vorgehensweise zur sicheren Verwaltung der Berechtigungen

Die private Registry ermöglicht die sichere Verwaltung der privaten Berechtigungen einer Einheit, während die öffentliche Registry für die Verwaltung der öffentlichen Berechtigungen bestimmt ist.

Die private Registry stellt eine Basisregistrierungsdatenbank mit sicheren bzw. verschlüsselten Token bereit. Sie kann z. B. als gesichertes Repository für öffentliche Elemente wie Minizertifikate und für private Elemente wie private Schlüssel verwendet werden.

Die private Registry sorgt dafür, dass nur berechtigte Benutzer auf private Elemente zugreifen können. Normalerweise kann nur ein WS-Manager-Benutzer mit entsprechender Berechtigung über einen PIN-Code auf die Registry zugreifen. Mit Hilfe der Konfigurationsoptionen können Sie dies jedoch umgehen, sofern keine allzu großen Sicherheitsbedenken bestehen.

Die private Registry stellt außerdem Unterstützung für Services (z. B. digitale Signaturen und RSA-Entschlüsselung) zur Verfügung, über die sichergestellt wird, dass die privaten Objekte nie diese private Datenbank verlassen. Über eine einheitliche Schnittstelle wird die zu Grunde liegende Einheitenunterstützung verdeckt; diese Unterstützung ist momentan auf das lokale Dateisystem beschränkt.

Automatische Registrierung: MQe bietet Standardservices zur Unterstützung der automatischen Registrierung. Diese Services werden bei der Konfiguration einer authentifizierbaren Einheit automatisch ausgelöst, beispielsweise beim Start eines WS-Managers oder bei der Definition einer neuen Warteschlange. In beiden Fällen werden die Registrierung gestartet und neue Berechtigungen erstellt und in der privaten Registry der Einheit abgelegt. Daher können mit der automatischen Registrierung auf einfache Weise Berechtigungen für den Schutz auf Nachrichtenebene eingerichtet werden.

Zur automatischen Registrierung gehören unter anderem folgende Schritte:

1. Generierung eines neuen RSA-Schlüsselpaars
2. Schutz und Speicherung des privaten Schlüssels in der privaten Registry
3. Verpackung des öffentlichen Schlüssels in die Anforderung eines *neuen Zertifikats* vom Standard-Minizertifikatsserver

Ist der Minizertifikatsserver konfiguriert und verfügbar, gibt er das neue Minizertifikat für die Einheit zusammen mit dem eigenen zurück. Diese werden zusammen mit dem geschützten privaten Schlüssel als neue Berechtigungen der Einheit in deren privater Registry gespeichert.

Öffentliche Registry und Zertifikatreplikation: MQe bietet Standardservices, mit denen MQe-Komponenten Minizertifikate verwenden können. Die öffentliche MQe-Registry stellt ein Repository mit öffentlichen Zugriffsmöglichkeiten für Minizertifikate dar. Dies entspricht einem persönlichen Telefonverzeichnis auf einem Mobiltelefon, nur dass es sich hier nicht um Telefonnummern, sondern um die Minizertifikate der authentifizierbaren Einheiten handelt, zu denen am häufigsten eine Verbindung hergestellt wird.

Bei den Services der öffentlichen Registry handelt es sich nicht nur um passive Services. Wird die Datenbank nach einem nicht vorhandenen Minizertifikat durchsucht und ist sie mit einer gültigen Ausgangsserverkomponente konfiguriert, ruft die öffentliche Registry das angeforderte Minizertifikat automatisch aus der öffentlichen Registry des Ausgangsservers ab. Diese Services können als intelligenter Service für die automatische Replikation von Minizertifikaten eingesetzt werden, der dafür sorgt, dass immer die richtigen Minizertifikate zur Verfügung stehen.

Verwendung der Registryservices durch Anwendungen: Der MQe-WS-Manager nutzt die Vorteile der privaten und öffentlichen Registrys, allerdings ist der Zugriff auf diese Services nicht eingeschränkt. MQe-Lösungen können eigene Einheiten (wie z. B. Benutzer) definieren und verwalten. Die Services der privaten Registry können dann für die automatische Registrierung und die Verwaltung der Berechtigun-

gen der neuen Einheiten verwendet werden, während die Services der öffentlichen Registry den Zugriff auf öffentliche Berechtigungen ermöglichen, wann immer dies erforderlich ist.

Service zur Standardausgabe von Minizertifikaten: Das SupportPac *MQe Server Support* enthält einen WTLS-Minizertifikatsserver und kann separat kostenlos heruntergeladen werden.

Das Softwarepaket enthält einen Service für die Vergabe von WTLS-Zertifikaten. Auf diesem Server für die Zertifikatsvergabe können WS-Manager und Warteschlangeneinheiten für einen Service zur Vergabe standardmäßiger Minizertifikate konfiguriert werden. Die Anforderungen der automatischen Registrierung für die private Registry erfüllt dieser Service dann durch Vergabe von WTLS-Zertifikaten. Mit den von MQe zur Verfügung gestellten Möglichkeiten kann ein Service für die Vergabe von Minizertifikaten an sorgfältig ausgewählte Einheitenamen eingerichtet und verwaltet werden. Dieser Service verfügt über folgende Merkmale:

- Verwaltung einer Gruppe registrierter authentifizierbarer Einheiten
- Vergabe von Minizertifikaten
- Verwaltung eines Repositories für WAP-WTLS-Minizertifikate

Sicherheitsschnittstelle: Es steht eine optionale Schnittstelle zur Verfügung, die von einem benutzerdefinierten Sicherheitsmanager implementiert werden kann. Die Methoden ermöglichen dem Sicherheitsmanager, folgende Anforderungen zuzulassen bzw. zurückzuweisen:

- Hinzufügen und Entfernen von Klassenaliasnamen
- Definieren von Adaptern
- Zuordnen von Dateideskriptoren
- Verarbeiten von Verbindungsbefehlen

Regeln für benutzerdefinierte Konfiguration

Mit Hilfe von Regeln können Benutzer das Verhalten einiger der Hauptkomponenten von MQe anpassen.

Wo dies notwendig ist, stellt MQe Standardregeln zur Verfügung, die jedoch nach Bedarf durch anwendungs- oder installationsspezifische Regeln ersetzt werden können, um den Kundenanforderungen gerecht zu werden.

Die unterstützten Regelarten unterscheiden sich sowohl durch die Auslöseart als auch durch ihre Funktion.

Regeln enthalten Logik und können daher eine Vielzahl von Funktionen ausführen.

Attributregeln: Attributregeln gelten nur für die Java-Codebasis.

An diese Regelklasse wird die Steuerung übergeben, sobald eine Statusänderung vorgenommen werden soll, wie beispielsweise eine Änderung der:

- Authentifizierungsfunktion
- Komprimierungsfunktion
- Verschlüsselungsfunktion

Die Regel kann die Änderung akzeptieren oder zurückweisen.

MQ-Brückenregeln: MQ-Brückenregeln betreffen nur die Java-Codebasis.

Diese Regelklassen werden aufgerufen, sooft bei der Ausführung des Codes der Brücke von MQe zu MQ eine Statusänderung auftritt. Zur Ermittlung der folgenden Informationen dient jeweils eine eigene Brückenregelklasse:

- Vorgehensweise, wenn eine von MQ gesendete Nachricht nicht von der Empfangsfunktion an MQe übergeben werden kann, z. B. weil die Nachricht zu groß oder die Warteschlange nicht vorhanden ist.

- Der Status, in dem von der Brücke verwaltete Warteschlangen unmittelbar nach dem Erstellen einer Serverinstanz gestartet werden sollen.
- Vorgehensweise, wenn die Brücke ein Problem in Zusammenhang mit der Sync-Warteschlange von MQ (dem permanenten Speicher, der für die Wiederherstellung nach einem Absturz herangezogen wird) feststellt (die Standardregel zeigt nur den Fehler an).
- Vorgehensweise zur Konvertierung einer MQE-Nachricht in eine MQ-Nachricht und umgekehrt, mit Hilfe von Transformatoren

Warteschlangenregeln: Diese Regelklasse wird an den Schlüsselstellen im Lebenszyklus einer Warteschlange aufgerufen, z. B. in folgenden Fällen:

- Eine Nachricht wird in eine Warteschlange eingereiht, um festzustellen, ob ein Schwellenwert überschritten ist (d. h. die Anzahl der Nachrichten oder Nachrichtenlänge überschritten worden ist).
- Eine Warteschlange wird geöffnet bzw. geschlossen.
- Eine Warteschlange wird von einem Warteschlangenmanager entfernt. Dies gilt nicht für die native C-Codebasis.
- Eine Nachricht in einer Warteschlange hat das Ablaufdatum der Warteschlange oder ihr eigenes Ablaufdatum überschritten.

WS-Managerregeln: Diese Regelklasse wird an den Schlüsselstellen im Lebenszyklus eines Warteschlangenmanagers aufgerufen, z. B. in folgenden Fällen:

- Ein WS-Manager wird geöffnet, und es kann beispielsweise ein Zeitgeber-Thread, der im Hintergrund abläuft, gestartet werden, damit zeitlich angesetzte Vorgänge aktiviert werden.
- Ein WS-Manager wird geschlossen, und es kann beispielsweise ein im Hintergrund ablaufender Zeitgeber-Thread beendet werden.
- Die Übertragung der Nachrichten, die auf dem WS-Manager zur Übertragung anstehen, wird ausgelöst.

Klassen

Dieser Abschnitt gilt nicht für die native C-Codebasis von MQE.

MQe bietet eine Auswahl an Klassen für bestimmte Funktionen, mit denen das Verhalten von MQe spezifischen Anwendungsanforderungen angepasst werden kann. In einigen Fällen werden die Schnittstellen zu Klassen dokumentiert, so dass die Entwicklung zusätzlicher Lösungen ermöglicht wird. Die untenstehende Tabelle fasst die Möglichkeiten zusammen. Klassen können entweder direkt oder über Aliasnamen identifiziert werden.

Anmerkung: Einige der Klassen stehen nicht in der C-Bindings-API zur Verfügung. Verbindliche Listen mit den unterstützten Klassen finden Sie im Handbuch 'Java Programming Reference' sowie im Handbuch 'C Programming Reference'.

Viele dieser Klassen werden von MQe automatisch mit einem Aliasnamen versehen. Diese sind im Handbuch 'Java Programming Reference' in 'com.ibm.mqe.MQe.alias' dokumentiert.

Tabelle 4. Klassenoptionen

Klasse	Unterstützte Alternativen	Dokumentierte Schnittstellen	MQe-Paket	Implementierungsansatz
Verwaltung	Nein	Ja		
Authentifizierungsfunktionen	Ja	Nein	com.ibm.mqe.attributes	com.ibm.mqe.MQeAuthenticator erweitern
Kommunikationsadapter	Ja	Ja	com.ibm.mqe.adapters	com.ibm.mqe.adapters.MQeCommunicationsAdapter erweitern

Tabelle 4. Klassenoptionen (Forts.)

Klasse	Unterstützte Alternativen	Dokumentierte Schnittstellen	MQe-Paket	Implementierungsansatz
Übertragungsarten	Ja	Nein		
Komprimierungsfunktionen	Ja	Nein	com.ibm.mqe.attributes	com.ibm.mqe.MQeCompressor erweitern
Verschlüsselungsfunktionen	Ja	Nein	com.ibm.mqe.attributes	com.ibm.mqe.MQeCryptor erweitern
Ereignisprotokoll	Beispiel verfügbar	Ja		com.ibm.mqe.MQeEventLogInterface implementieren
Nachrichten	Nein	Ja	com.ibm.mqe	com.ibm.mqe.MQeMsgObject erweitern
Warteschlangenspeicher	Ja	Nein		Normalerweise sollte der vom Aliasnamen MsgLog: definierte Standard verwendet werden.
Regeln	Standardklassen verfügbar	Ja		com.ibm.mqe.MQeRule erweitern
Speicheradapter	Ja	Ja	com.ibm.mqe.adapters	com.ibm.mqe.adapters.MQeAdapter erweitern
Trace	Beispiele verfügbar	Ja	com.ibm.mqe.trace	

Anwendungen laden:

Anmerkung: Dieser Abschnitt gilt nicht für die C-Codebasis.

Wenn ein MQe-WS-Manager geladen wird, muss die Anwendung, die diesen Vorgang eingeleitet hat, auch alle anderen Anwendungen in die JVM laden.

Hierzu können entweder standardmäßige Java-Funktionen oder das Klassenladeprogramm, das Bestandteil von MQe ist, verwendet werden.

Daher gilt:

- Es können mehrere Anwendungen mit einem einzigen Warteschlangenmanager in derselben JVM ausgeführt werden.
- Ebenso können mehrere JVMs verwendet werden, allerdings ist für jede ein eigener WS-Manager mit einem eindeutigen Namen erforderlich.

MQe-SupportPacs

Bei MQe handelt es sich um eine Produktfamilie, die Tools für die Entwicklung, den Einsatz und die Verwaltung von Messaging- und -Queuing-Lösungen mit MQe zur Verfügung stellt.

Die Produktfamilie umfasst Folgendes:

1. Das **Lizenzprodukt MQe**, das entweder auf einem physischen Datenträger von IBM erworben oder aus dem Web unter folgender Adresse heruntergeladen werden kann:

<http://www.ibm.com/software/integration/wmqe/>

Das Lizenzprodukt umfasst folgende Komponenten:

- MQe-Java-Klassen
- Hilfsklassen
- MQe-C-Bindings-Dateien und native C-Codebasis
- Beispielquellcode für Anwendungen
- Dienstprogramme
- Referenzhandbücher
- Lizenzinformationen

Das Lizenzprogramm selbst enthält auch die Berechtigung für die Verwendung des Produkts für Produktionszwecke (nicht Entwicklungszwecke) auf bestimmten Plattformen. Bei großen Systemen oder falls die MQ-Brücke eingesetzt werden soll, müssen unter Umständen weitere Kapazitätseinheiten erworben werden.

2. MQe-SupportPacs, zum Download unter folgender Web-Adresse verfügbar:

<http://www.ibm.com/software/integration/support/supportpacs/>

bzw.

<http://www.ibm.com/software/integration/wmqe/>

Die Verwaltungstools in den MQe-SupportPacs spielen eine wichtige Rolle in der Entwicklung, Installation und Konfiguration von Anwendungen. Sie bieten wesentlich mehr Funktionen als die im Lizenzprogramm verfügbaren Dienstprogramme und sind unerlässlich für die Einführung, Konfiguration und den Test von Pilotnetzen sowie für die Verwaltung von Produktionssystemen.

EA01: WebSphere MQ Everyplace - XML-Konvertierungsdienstprogramm

Software, mit der sich ein MQeFields-Objekt in seine XML-Darstellung konvertieren lässt und umgekehrt.

EP02: WebSphere MQ Everyplace - JDBC Adapter

Erweitert die permanente Speicherung von Nachrichten auf den Bereich innerhalb von DB2-Datenbanken.

ES06: WebSphere MQ Everyplace Server Support

Fasst MQe_Explorer, MQe_Script, MQe_MinicertServer und MQe_Service zusammen, die Verwaltungs- und Sicherheitsfunktionen für MQe zur Verfügung stellen.

MS0B: MQSeries Java-Klassen für PCF

Java-Code zur Unterstützung von PCF-Nachrichten. Seine Verwendung wird im Abschnitt „MS0B - MQSeries-Java-Klassen für PCF“ erklärt.

MS0B - MQSeries-Java-Klassen für PCF

PCF-Nachrichten sind Verwaltungsnachrichten, die von MQ-Warteschlangenmanagern verwendet werden. Dieses SupportPac enthält Java-Code, der PCF-Nachrichtenunterstützung liefert.

Wenn Sie es herunterladen und installieren und die Datei `com.ibm.mq.pcf.jar` in die Klassenpfad-Umgebungsvariable aufnehmen, haben Sie Zugriff auf Java-Klassen, die MQ-Ressourcen dynamisch manipulieren können. Wenn PCF-Nachrichten mit MQe-Verwaltungsnachrichten kombiniert werden, ist die vollständige programmgestützte Konfiguration von Brückenressourcen und entsprechenden Ressourcen eines MQe-Warteschlangenmanagers möglich. In der Klasse `examples.mqbridge.administration.programming.AdminHelperMQ` enthaltener Beispielcode zeigt in Verbindung mit `examples.mqbridge.administration.programming.MQAgent`, wie das angegangen werden kann. Dieser Beispielcode ist in das Programm `examples.awt.AwtMQeServer` aufgenommen worden, so dass die Auswahl von **View (Anzeigen) → Connect local MQ default queue manager (Lokalen MQ-Standardwarteschlangenmanager verbinden)** Folgendes bewirkt:

- Sie stellt sicher, dass ein Brückenobjekt existiert, indem sie bei Bedarf eines erstellt.
- Sie fragt Eigenschaften beim Standard-MQ-Warteschlangenmanager ab.
- Sie versucht, diesen Warteschlangenmanager mit dem derzeit aktiven MQe-Warteschlangenmanager zu verbinden.

- Sie stellt sicher, dass ein Proxy-Objekt existiert, das den Standard-MQ-Warteschlangenmanager darstellt, indem sie bei Bedarf eines erstellt.
- Sie stellt sicher, dass eine MQe-Clientverbindung sowie ein entsprechender MQ-Serververbindungskanal existiert, indem sie diese Ressourcen bei Bedarf erstellt.
- Sie stellt sicher, dass auf dem MQ-Warteschlangenmanager eine *synchrone Warteschlange* vorhanden ist.
- Sie stellt sicher, dass in MQ eine Übertragungswarteschlange vorhanden ist, und erstellt bei Bedarf eine solche.
- Sie stellt sicher, dass eine dazu passende MQ-Übertragungswarteschlangen-Empfangsfunktion in der Konfiguration des aktuellen MQe-Warteschlangenmanagers vorhanden ist, indem sie bei Bedarf eine erstellt.
- Sie stellt sicher, dass alle Brückenressourcen gestartet sind.
- Sie stellt sicher, dass auf dem MQ-Warteschlangenmanager eine Test-Warteschlange vorhanden ist, indem sie bei Bedarf eine erstellt.
- Sie stellt sicher, dass eine dazu passende MQe-Brückenwarteschlange vorhanden ist, die auf diese Test-Warteschlange verweist.
- Sie sendet eine Testinstanz von MQeMQMMsgObject an die Test-Warteschlange, um sicherzustellen, dass die Konfiguration funktioniert.
- Sie ruft die Testinstanz von MQeMQMMsgObject aus der Test-Warteschlange ab, um sicherzustellen, dass die Konfiguration funktioniert.

Planung der Implementierung

Lizenzen

Zur Implementierung Ihrer MQe-Anwendungen erforderliche Lizenzen

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 hundert (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 im Abschnitt <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 im Abschnitt <http://www.ibm.com/software/integration/mqfamily/>.

Benötigte Hardware und Software

Zur Entwicklung und Implementierung von Anwendungen benötigte Hardware und Software

Sie benötigen:

- Einen PC zum Schreiben und Kompilieren Ihrer Anwendung.
Es wird ein Windows-System empfohlen, weil Sie dann jedes beliebige der MQe-SupportPacs ausführen können, insbesondere den MQe-Explorer, der bei der Entwicklung von großem Nutzen ist. Dieser Rechner sollte über einen Internet-Zugang verfügen, damit Sie MQe, SupportPacs, Dokumentation usw. herunterladen können.
- Mindestens einen der Rechner bzw. eins der Geräte, auf denen Sie die Implementierung beabsichtigen, zu Testzwecken
- Gegebenenfalls Schnittstellenhardware und Kabel zum Anschließen des Geräts an Ihren Entwicklungs-PC.

Zu verwendende Codebasis

Bei der MQe-Anwendungsprogrammierschnittstelle (API) handelt es sich um die Programmierschnittstelle zu MQe. Es werden zwei Sprachen unterstützt, Java und C.

Die Java-Version bietet Zugriff auf alle MQe-Funktionen. Die Klassen, Methoden und Prozeduren im Einzelnen sind in dem Dokument Java Programming Reference beschrieben. An zahlreichen Stellen in diesem Information Center finden sich Beispiele zur MQe-Programmierung.

Es gibt drei Versionen der C-Unterstützung:

Die native C-Codebasis bietet Zugriff auf einen bedeutenden Teil der MQe-Funktionen. Da es sich bei der C-Codebasis nur um einen Geräte-Warteschlangemanager handelt, bestehen folgende Einschränkungen:

- Keine Unterstützung von Warteschlangen zum Speichern und Weiterleiten sowie von Warteschlangen mit Brückenfunktion
- Unterstützt nur den HTTP-Adapter
- Unterstützt nur die RLE-Komprimierungsfunktion
- Unterstützt nur die RC4-Verschlüsselungsfunktion
- Unterstützt nur das Attribut *MAttribute* und die lokalen Sicherheitseinrichtungen

Die Methoden und Prozeduren im Einzelnen sind im Dokument C Programming Reference beschrieben. An zahlreichen Stellen in diesem Information Center finden sich Beispiele zur MQe-Programmierung für die C-Bindings.

Die C-Bindings werden für die Verwendung bis zur Verfügbarkeit des vollen Funktionsumfangs in der nativen C-Codebasis bereitgestellt. Sie bietet Zugriff auf einen bedeutenden Teil der MQe-Funktionen. Die Methoden und Prozeduren im Einzelnen sind im Dokument C Programming Reference beschrieben. Beispiele zur MQe-Programmierung für die C-Bindings finden sich im Dokument C Bindings Programming Reference.

MQe-Entwicklungszyklus

In Anbetracht des breiten Spektrums von Einsatzmöglichkeiten für MQe wird das Produkt nicht so installiert, konfiguriert und implementiert wie andere Produkte der MQ-Produktfamilie. Die Einrichtung von MQe wird in drei Phasen vollzogen:

1. Phase der Entwicklung und Erstellung von Prototypen

MQe steht im Rahmen der Bedingungen der MQe-Entwicklungslizenz zur kostenlosen Installation und Verwendung zur Verfügung. MQe-Anwendungen werden unter Verwendung der Funktionen entwickelt, die die JMS-API bzw. die MQe-Java-API und die C-API zur Verfügung stellen. Hinsichtlich der Erstellung und Verwendung eines MQe-Warteschlangenmanagers sind die im Folgenden besprochenen Gesichtspunkte zu beachten. Diese Sachverhalte können Auswirkungen darauf haben, wie Sie Ihre Anwendung implementieren.

- Es kann zu jedem Zeitpunkt jeweils nur ein MQe-Warteschlangenmanager in einer JVM bzw. einem C-Prozess aktiv sein. In MQe wird eine Prüfung durchgeführt, und es wird ein Fehler ausgelöst, wenn eine Anwendung versucht, mehr als einen Warteschlangenmanager zu starten. Wenn mehrere Anwendungen einen einzigen Warteschlangenmanager verwenden muss, müssen diese in derselben JVM angesiedelt sein. Wenn mehrere Anwendungen in derselben JVM ausgeführt werden, können die Anwendungen die Methode `MQeQueueManager.getDefaultQueueManager()` verwenden, um zu überprüfen, ob bereits ein Warteschlangenmanager ausgeführt wird. Auf einer einzigen Maschine können beliebig viele Anwendungen und Warteschlangenmanager verwendet werden, allerdings ist mit Blick auf die Geschwindigkeit und die Beanspruchung der Systemressourcen Vorsicht angebracht.
- Pro Nachrichtenspeicher sollte nur ein MQe-Warteschlangenmanager gestartet werden. Bei Verwendung einer grafischen Benutzeroberfläche ist es relativ einfach, mehr als eine Instanz einer Anwendung zu starten. Daher ist es möglich, mehrere Instanzen desselben Warteschlangenmanagers über demselben Nachrichtenspeicher zu starten. Dies kann zu unklaren Ergebnissen führen, besonders bei Verwendung von asynchroner Warteschlangensteuerung.

C Die C-API bietet Zugriff auf die der Rolle des Client-Warteschlangenmanagers entsprechenden Funktionen. Die .NET-Umgebung kann verwendet werden, indem die Plattform Invocation Services zu Aufrufen an die MQe-C-DLLs verwendet werden. Die erforderlichen DLLs müssen auf das Gerät heruntergeladen werden.

Es ist äußerst ratsam, während der Phase der Prototyperstellung Tests über dasjenige Netzwerk auszuführen, das später im laufenden Betrieb verwendet werden soll, und außerdem Daten zu verwenden, wie sie später auch im laufenden Betrieb verwendet werden. Dadurch können Sie zu zutreffenden Erwartungen zur Leistungsstärke gelangen und die Standardübertragungseinstellungen in MQe richtig einschätzen.

Die Entwicklungslizenz beinhaltet keine Unterstützung durch IBM. Bei der Implementierungslizenz (siehe unten) dagegen wird Unterstützung während der Anwendungsentwicklung und darüber hinaus geboten.

2. Einsatzphase

In dieser Phase findet die Verwendung der entwickelten Anwendungen statt; daher sind entsprechend den Bedingungen der MQe-Lizenz Kapazitätseinheiten für die Verwendung des Produkts erforderlich. Die Java-Klassen und die C-API dürfen nur mit Zustimmung von IBM mit der Anwendung weitergegeben werden, oder wenn die Benutzer bereits die Berechtigung zu ihrer Verwendung haben. Andernfalls müssen die Benutzer in Java die erforderlichen Klassen selbst anpassen und in C MQe auf das Gerät kopieren.

3. Verwaltungsphase

Wenn die MQe-WS-Manager im Netz aktiv sind, sind Tools für ihre Überwachung und Verwaltung erforderlich. Unterstützung für MQe ist im Rahmen der IPLA-Lizenzbedingungen gewährleistet.

Unterstützungsstufen

Auf Grund dieses Einsatzkonzeptes werden auch Unterschiede bei der Plattformunterstützung gemacht. Bei MQe mit Kapazitätseinheiten und SupportPacs der Kategorie 3 unterscheidet IBM zwischen den folgenden Plattformen:

- Plattformen, auf denen die Installation und Anwendungsentwicklung unterstützt wird:
 - Fehlerberichte zu Problemen mit der Installation, Anwendungsentwicklung und Verwendung können eingeschickt werden.
- Plattformen, auf denen der Einsatz von Anwendungen zulässig ist, jedoch nicht direkt unterstützt wird:
 - Es kann verlangt werden, dass die in Fehlerberichten geschilderten Probleme auf einer unterstützten Plattform reproduziert werden.
- Plattformen, auf denen der Einsatz von Anwendungen unterstützt wird:
 - Fehlerberichte zum Einsatz von Anwendungen können eingeschickt werden.

Einstieg in MQe

Es gibt verschiedene Möglichkeiten, sich in MQe einzuarbeiten.

- Einen Warteschlangenmanager betriebsbereit zu machen und anschließend ein einfaches MQe-Netzwerk einzurichten, ist eine produktive Methode, sich mit dem Produkt und seinen Konzepten vertraut zu machen.
- Auf der Basis dieser Kenntnisse können Sie sich dann mit der Erstellung einfacher Anwendungen weiter in das Produkt einarbeiten.
- Am Anfang ist es im Allgemeinen nicht hilfreich, sich mit anderen Produkten der MQ-Produktfamilie zu befassen. Später jedoch, wenn die Brückenfunktion relevant wird, ist es wichtig, dies zu verstehen.

Im Sinne dieser Strategie wird neuen Benutzern empfohlen, sich mit den in diesem einführenden Teil der Dokumentation vorgestellten Grundbegriffen vertraut zu machen.

Wenn Sie Zugang zu einem System mit einem Windows-Betriebssystem haben, laden Sie SupportPac ES06, "MQe Server Support", herunter, das MQe_Explorer enthält, und folgen Sie den Anweisungen für die ersten Schritte mit MQe. Die Installation des Lizenzprodukts ist nicht unbedingt erforderlich, allerdings gelten ohne dieses Produkt die in der Lizenzvereinbarung genannten Einschränkungen.

Verwendung von MQe mit MQ

Einführung

Ein MQe-Netz kann zwar als Standalone-Installation, d. h. ohne MQ-Server oder -Netz betrieben werden, in der Praxis wird es aber häufig als Ergänzung zu einer vorhandenen MQ-Installation eingesetzt.

Damit wird deren Reichweite auf neue Plattformen und Geräte ausgedehnt, und es kommen fortgeschrittene Leistungsmerkmale hinzu, wie beispielsweise warteschlangen- oder nachrichtenbasierte Sicherheit und synchrone Nachrichtenübertragung.

Aus Sicht einer MQe-Anwendung fungieren MQ-Warteschlangen und -Warteschlangenmanager als zusätzliche ferne Warteschlangen und Warteschlangenmanager. Allerdings bestehen hier einige funktionale Einschränkungen, da der Zugriff auf diese Warteschlangen nicht direkt über MQe-Verbindungen und einen MQe-WS-Manager erfolgen kann, sondern ein MQe-Gateway erfordert.

Das Gateway kann Nachrichten an mehrere Warteschlangenmanager senden, entweder direkt oder indirekt über MQ-Clientkanäle. Bei einer indirekten Verbindung werden die Nachrichten über MQ-Clientkanäle an zwischengeschaltete MQ-WS-Manager gesendet und von dort über MQ-Nachrichtenkanäle an den Ziel-WS-Manager übertragen.

Gateway (Brücke) zu MQ

Dieser Abschnitt gilt nicht für die C-Codebasis.

MQe unterstützt die *MQ-Brücke*, die als Schnittstelle zwischen MQe- und MQ-Netzwerken fungiert.

Diese Brücke stellt über den MQ-Java-Client eine Verbindung zu einem oder mehreren MQ-WS-Managern her und ermöglicht damit die Übermittlung von Nachrichten von MQe an MQ und umgekehrt.

In der aktuellen Version von MQe gilt:

- Es wird eine solche Brücke pro Server empfohlen.
- Jede Brücke ist mit mehreren *MQ-Warteschlangenmanager-Proxys* (Definitionen von MQ-Warteschlangenmanagern) verbunden.
- Für jeden MQ-Warteschlangenmanager, der mit MQe kommuniziert, ist eine *Warteschlangenmanager-Proxy-Definition* erforderlich.
- Jeder dieser Definitionen können ein oder mehrere *Client-Verbindungsservices* zugeordnet werden, wobei jeder die Verbindung zu einem einzelnen MQ-WS-Manager darstellt.
- Diese Client-Verbindungsservices können jeweils unterschiedliche MQ-Serververbindungen zum WS-Manager sowie unterschiedliche Merkmale wie beispielsweise Benutzer-Exits oder Anschlüsse verwenden.

Nachrichtenkonvertierung

Die an MQ gerichteten MQe-Nachrichten werden über die Brücke gesendet und mit Hilfe eines Standardtransformators bzw. des Transformators der Zielwarteschlange in ein MQ-Format konvertiert. Benutzerdefinierte Transformatoren sind sehr flexibel; beispielsweise ist es sinnvoll, eine Unterklasse der MQe-Nachrichtenklasse für die Darstellung einer bestimmten Nachrichtenart im gesamten MQe-Netz zu verwenden. Auf dem Gateway kann die Nachricht von einem Transformator unter Zuordnung der entsprechenden Felder und MQ-Werte in ein MQ-Format konvertiert werden, wobei bestimmte Daten hinzugefügt werden, die die Bedeutung der Unterklasse angeben.

Der Standardtransformator für die Konvertierung vom MQe- in das MQ-Format kann diese Unterklasseninformationen zwar nicht verwenden, ist jedoch trotzdem vielseitig einsetzbar. Er weist folgende Merkmale auf:

• Nachrichtenfluss von MQe an MQ:

Der Standardtransformator für die Konvertierung vom MQe- in ein MQ-Format funktioniert in Verbindung mit der Klasse *MQeMQMsgObject*. Diese Klasse stellt alle Felder dar, die möglicherweise im MQ-Nachrichtenheader vorhanden sind.

Mit Hilfe dieser Klasse kann Ihre Anwendung unter Verwendung der Methode *'set()'* die erforderlichen Werte setzen. Wird eine Instanz von *'MQeMQMsgObject'* bzw. eine Ableitung davon an den MQe-Standardtransformator (*MQeBaseTransformer*) übergeben, ruft dieser Transformator die Werte aus der Instanz von *'MQeMQMsgObject'* ab und setzt die entsprechenden Werte in der MQ-Nachricht; beispielsweise wird der Wert für die Priorität in die MQ-Nachricht kopiert.

Handelt es sich bei der Nachricht weder um ein *'MQeMQMsgObject'* noch um eine Ableitung davon, wird die gesamte MQe-Nachricht in den Hauptteil der MQ-Nachricht kopiert. Diese Vorgehensweise wird als *Funneling* bezeichnet. Der im Nachrichtenformatfeld des MQ-Nachrichtenheaders gesetzte Wert gibt an, dass die MQ-Nachricht eine Nachricht im *MQe-Funneling*-Format enthält.

• Nachrichtenfluss von MQ an MQe:

An MQe geleitete MQ-Nachrichten werden in ähnlicher Weise behandelt wie die in umgekehrter Richtung geleiteten. Der Standardtransformator überprüft das Nachrichtenartfeld im MQ-Nachrichtenheader und geht entsprechend vor.

Enthält der MQ-Nachrichtenheader den Hinweis, dass es sich um eine MQe-Nachricht im *Funneling*-Format handelt, wird aus dem MQ-Nachrichtenhauptteil die ursprüngliche MQe-Nachricht wiederhergestellt und an das MQe-Netz gesendet.

Handelt es sich nicht um eine MQe-Nachricht im *Funneling*-Format, wird der Inhalt des MQ-Nachrichtenheaders extrahiert und in ein 'MQeMQMsgObject' gepackt. Der MQ-Nachrichtenhauptteil wird wie ein einfaches Bytefeld behandelt und ebenfalls in das 'MQeMQMsgObject' kopiert. Anschließend wird dieses Objekt an das MQe-Netz gesendet.

Die Klasse 'MQeMQMsgObject' und das Standardverhalten des Transformators bedeuten Folgendes:

- Eine MQe-Nachricht kann unverändert durch ein MQ-Netzwerk an ein MQe-Netzwerk gelangen.
- Eine MQ-Nachricht kann unverändert durch ein MQe-Netzwerk an ein MQ-Netzwerk gelangen.
- Eine MQe-Anwendung kann eine beliebige MQ-Anwendung steuern, ohne dass die MQ-Anwendung geändert wird.

Funktion

Ferne MQ-Warteschlangen ermöglichen das synchrone Einreihen (Put) von MQe-Nachrichten von einem MQe-WS-Manager aus.

Alle anderen Vorgänge im Zusammenhang mit der Nachrichtenübertragung müssen asynchron erfolgen.

MQe-Verwaltungsnachrichten können nicht an einen MQ-WS-Manager gesendet werden. Die Verwaltungswarteschlange ist dort nicht vorhanden, und das Format der Verwaltungsnachrichten unterscheidet sich von dem, das von MQ verwendet wird.

Kompatibilität

Ein MQe-Netz kann zwar ohne MQ eingesetzt werden, in vielen Fällen sind jedoch für die Anwendungsausführung beide Produkte erforderlich. MQe kann in ein vorhandenes MQ-Netz integriert werden, wobei Kompatibilität in folgenden Punkten gewährleistet sein muss:

Adressierung und Benennung:

- Identische Adressierungssemantik unter Verwendung einer WS-Manager- oder Warteschlangenadresse
- Einheitliche Verwendung eines ASCII-Namensbereichs

Anwendungen:

- MQe unterstützt vorhandene MQ-Anwendungen, ohne dass hierzu Änderungen erforderlich sind.

Verbindungen:

- Das MQe-Gateway verwendet MQ-Clientkanäle.

Nachrichtenaustausch und -inhalt:

- Nachrichtenaustausch zwischen MQe und MQ
- Intransparentes Nachrichtennetz (MQe- bzw. MQ-Nachrichten können ohne Änderungen über das jeweils andere Netz geleitet werden)
- Gegenseitige Unterstützung gekennzeichnete Felder im MQ-Nachrichtenheader
- Einmalige gesicherte Nachrichtenzustellung

MQe unterstützt nicht alle Funktionen von MQ. Einige der wichtigsten Unterschiede (neben umgebungs-, betriebssystem- und übertragungsspezifischen Überlegungen) sind nachfolgend aufgeführt:

- Keine Clusterunterstützung
- Keine Unterstützung von Verteilerlisten
- Keine gruppierten oder segmentierten Nachrichten
- Keine Möglichkeiten zum Lastausgleich bzw. zum Einsatz eines Ersatzsystems (Warm Standby)
- Keine Referenznachrichten
- Keine Berichtsoptionen
- Keine Unterstützung für gemeinsame Warteschlangen
- Keine Auslösefunktion

- Keine UOW-Unterstützung (UOW = Unit of Work; Arbeitseinheit), keine XA-Koordinierung
- Unterschiede bei der Skalierbarkeit und den Leistungskennwerten

Allerdings können viele Anwendungstasks in MQE mit alternativen Mitteln bewältigt werden, die MQE-Funktionen verwenden, oder durch die Ausnutzung der Möglichkeit zur Unterklassenbildung, die Ersetzung vorgegebener Klassen oder die Ausnutzung der Regeln, Schnittstellen und sonstigen Anpassungsmöglichkeiten, die in das Produkt eingebaut sind.

Garantierte Zustellung

Zwar bieten sowohl MQE als auch MQ die garantierte Zustellung, aber mit unterschiedlichen Garantiestufen.

- Die Übertragung einer Nachricht von MQE an MQ ist nur dann garantiert, wenn die Kombination von *putMessage* und *confirmPutMessage* verwendet wird.
- Die Übertragung einer Nachricht von MQE an MQ ist nur dann garantiert, wenn die MQ-Nachricht als persistent definiert ist.

Weitere Informationen

Referenzinformationen zu MQ

Nachfolgend sind verwandte MQ-Veröffentlichungen aufgeführt, die für Sie von Nutzen sein könnten:

WebSphere MQ: An Introduction to Messaging and Queuing (GC33-0805)

Dieses Buch beschreibt kurz, was MQ ist und wie es einige klassische Interoperabilitätsprobleme lösen kann.

WebSphere MQ-Einstiegshandbücher

Zu jeder von MQ unterstützten Plattform gibt es MQ-Einstiegshandbücher. Diese Bücher enthalten plattformspezifische Planungs- und Installationsinformationen zu MQ.

Websites

Die MQE-Homepage hat folgende Adresse:

<http://www.ibm.com/software/integration/wmq/>

Über die Links auf dieser Homepage können Sie:

- mehr über die Produktmerkmale und Vorzüge von MQE herausfinden
- Informationen über Schulungen und Zertifizierung erhalten
- auf die MQE-Handbücher in den Formaten PDF und HTML zugreifen
- die neuesten Upgrades und zu testenden Codefassungen herunterladen.

Die MQE-SupportPacs können Sie herunterladen, indem Sie auf der folgenden Seite das Produkt *WebSphere MQ Everywhere* auswählen:

<http://www.ibm.com/software/integration/support/supportpacs/>

Möglicherweise ist für Sie auch die Homepage von MQ von Interesse, die Sie unter folgender Adresse finden:

<http://www.ibm.com/software/integration/wmq/>

Gleiches gilt für die Homepage der MQ-Produktfamilie:

<http://www.ibm.com/software/integration/mqfamily/>

Auf die Bibliothek der Handbücher zur MQ-Produktfamilie können Sie unter folgender Adresse zugreifen:

<http://www.ibm.com/software/integration/websphere/library/books/>

Übersetzte Dokumentation

Das Einführungshandbuch zu MQe liegt auch in anderen Sprachen als Englisch vor. Diese übersetzten Dokumente können von der Website der MQ-Bibliothek unter folgender Adresse heruntergeladen werden:

<http://www.ibm.com/software/integration/websphere/library/>.

Newsgroups

Diese Newsgroups sind alle auf news.software.ibm.com und vielen anderen allgemein zugänglichen Newsservern zu finden.

Zu MQe:

- ibm.software.websphere.mqeveryplace

Zu MQ:

- ibm.software.websphere.mq
- ibm.software.websphere.mq.administration
- ibm.software.websphere.mq.programming

Weitere Fundstellen zum Thema:

- ibm.software.websphere.mqintegrator
- ibm.software.websphere.studio
- ibm.software.websphere.studio.verschiedene

MQe-Zertifizierung

Es bestehen Möglichkeiten, Schulungen zu MQe zu absolvieren und Zertifikate zu erwerben. Weitere Einzelheiten dazu finden Sie ausgehend von dieser Website:

<http://www.ibm.com/software/integration/websphere/education/>

Bemerkungen und Marken

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte oder IBM Services bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. An Stelle der Produkte, Programme oder Services können auch andere ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe
Director of Licensing
92066 Paris La Defense Cedex
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekannt gegeben. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire
England
SO21 2JN

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung sowie der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Marken

Die folgenden Namen sind in gewissen Ländern Marken der International Business Machines Corporation:

AIX, Everyplace, IBM, IBMLink, iSeries, MQSeries, SupportPac, WebSphere, z/OS, zSeries

Microsoft, Windows, Windows NT und das Windows-Logo sind in gewissen Ländern Marken der Microsoft Corporation.

Java und alle Java-basierten Marken und Logos sind in gewissen Ländern Marken oder eingetragene Marken von Sun Microsystems, Inc.

Linux ist in gewissen Ländern eine Marke von Linus Torvalds.

Andere Namen von Unternehmen, Produkten und Services können Marken oder Servicemarken anderer Unternehmen sein.

Glossar

In diesem Glossar werden die im vorliegenden Handbuch verwendeten Begriffe erläutert, sowie Begriffe, die hier in einer anderen als der sonst üblichen Bedeutung verwendet werden. In einigen Fällen gibt es für einen Begriff mehrere Definitionen; es wird jedoch nur jeweils die für dieses Handbuch zutreffende Definition angegeben.

Falls ein gesuchter Begriff nicht im Glossar enthalten ist, versuchen Sie es mit einer Softcopy-Suche oder konsultieren Sie den gedruckten Index oder das Buch *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

A
B
C
E
F
G
H
I
J
K
L
M
N
O
P
R
S
T
U
V
W

A

Anwendungsprogrammierschnittstelle (API)

Eine Anwendungsprogrammierschnittstelle (API; Application Programming Interface) setzt sich aus den Funktionen und Variablen zusammen, die von Programmierern in den von ihnen entwickelten Anwendungen verwendet werden können.

Asynchroner Nachrichtenaustausch

Ein Verfahren zur Kommunikation zwischen Programmen, bei dem Nachrichten in Nachrichtenwarteschlangen gestellt werden. Bei der asynchronen Nachrichtenübertragung setzen die Programme die Verarbeitung fort, ohne auf eine Antwort auf eine Nachricht zu warten. Vgl. Synchrone Nachrichtenübertragung.

Authentifizierungsfunktion

Ein Programm, das Sender und Empfänger von Nachrichten überprüft.

B

Brücke

Eine Komponente, die zu einem MQe-Warteschlangenmanager hinzugefügt werden kann, damit er mit MQ kommunizieren kann. Siehe MQe-Warteschlangenmanager.

C

Client In MQ ist ein Client eine Laufzeitkomponente, die es lokalen Benutzeranwendungen ermöglicht, Nachrichten an einen Server zu senden.

E

Einreihung von Nachrichten in Warteschlangen

Eine Programmiertechnik, bei der ein Programm einer Anwendung mit den anderen Programmen kommuniziert, indem Nachrichten in Warteschlangen eingereiht werden.

F

Ferner Warteschlangenmanager

Dieser Begriff wird im Zusammenhang mit der Definition einer fernen Warteschlange verwendet. Er beschreibt den fernen Warteschlangenmanager, dem die lokale Warteschlange gehört, die das Ziel der Definition einer fernen Warteschlange darstellt.

G

Gateway

Ein Rechner beliebiger Größe, auf dem ein MQe-Gatewaywarteschlangenmanager ausgeführt wird, der die MQ-Brückenfunktion beinhaltet. Siehe MQe-Warteschlangenmanager.

Gatewaywarteschlange

Ein Warteschlangenmanager mit einer Empfangsfunktion und einer Brücke. Siehe MQe-Warteschlangenmanager.

Geräteplattform

Ein Kleinrechner, auf dem MQe nur als Client ausgeführt werden kann, d. h. nur mit einem Geräte-Warteschlangenmanager.

Geräte-Warteschlangenmanager

Siehe MQe-Warteschlangenmanager.

Geräte-Warteschlangenmanager

Bei MQe: Ein Warteschlangenmanager ohne Empfangsfunktionskomponente und ohne Brückenkomponente. Er kann daher lediglich Nachrichten senden, aber keine empfangen.

Gatewaywarteschlange

Bei MQe: Ein Warteschlangenmanager, zu dem eine Empfangsfunktion und eine Brücke hinzugefügt werden können. Mit der Empfangsfunktion kann er auch Nachrichten empfangen, nicht nur senden, und mit der Brücke kann er mit MQ kommunizieren.

Geschützt

Ein geschütztes Feld ist nur innerhalb der eigenen Klasse, einer Unterklasse bzw. in Paketen, die die Klasse enthalten, sichtbar.

H

Hypertext Markup Language (HTML)

Eine Sprache zur Definition von Informationen, die im World Wide Web angezeigt werden sollen.

I

Instanz

Ein Objekt. Ein Objekt, das für eine Klasse angelegt wurde, wird als Instanz dieser Klasse bezeichnet.

Internet

Ein öffentliches Netz für die gemeinsame Nutzung von Informationen. Physisch gesehen greift das Internet auf einen Teil der Ressourcen zu, die über die derzeit vorhandenen öffentlichen Telekommunikationsnetze zur Verfügung stehen. In technischer Hinsicht zeichnet sich das Internet als gemeinschaftliches öffentliches Netz durch die Verwendung des TCP/IP-Protokolls (Transport Control Protocol/Internet Protocol) aus.

J

Java Development Kit (JDK)

Ein von Sun Microsystems für Java-Entwickler vertriebenes Softwarepaket. Es umfasst den Java-Interpreter, Java-Klassen sowie Java-Entwicklungstools: Compiler, Debugger, Disassembler, Appletviewer, Stub File Generator und Dokumentationsgenerator.

Java Naming and Directory Service (JNDI)

Eine in der Programmiersprache Java spezifizierte API. Es bietet in der Programmiersprache Java geschriebenen Anwendungen Benennungs- und Verzeichnisfunktionen an.

K

Kanal Siehe *Dynamischer Kanal* und *MQI-Kanal*.

Kanalmanager

Ein MQe-Objekt, das eine größere Anzahl gleichzeitiger logischer Übertragungsleitungen zwischen Endpunkten unterstützt.

Kapselung

Eine objektorientierte Programmieretechnik, bei der die Daten eines Objekts nicht öffentlich gemacht werden, also geschützt werden, und der Zugriff auf diese Daten und ihre Manipulation für Programmierer nur über Methodenaufrufe möglich ist.

Klasse Eine gekapselte Gruppe von Daten und Methoden, die zur Datenverarbeitung verwendet wird. Es besteht die Möglichkeit zur Erstellung von Klasseninstanzen, bei denen es sich um Objekte handelt.

Komprimierungsfunktion

Ein Programm zur Komprimierung von Nachrichten, das das zu übertragende Datenvolumen reduziert.

L

Lightweight Directory Access Protocol (LDAP)

Ein Client/Server-Protokoll für den Zugriff auf Verzeichnisservices.

M

Methode

In der objektorientierten Programmierung die Bezeichnung für eine Funktion oder Prozedur.

MQ-Brücke

Ein Rechner mit einem Gatewaywarteschlangenmanager, der mit MQ kommunizieren kann. Siehe MQe-Warteschlangenmanager.

MQ und MQ-Produktfamilie

Darunter ist **WebSphere MQ** zu verstehen, wozu folgende Produkte gehören:

- **WebSphere MQ Workflow** vereinfacht die unternehmensweite Integration durch die Automatisierung von Geschäftsprozessen, an denen Personen und Anwendungen beteiligt sind.
- **WebSphere MQ Integrator** ist eine Nachrichtenbroker-Software, die intelligentes, regelbasiertes Nachrichten-Routing in Echtzeit ermöglicht, sowie die Transformation und Formatierung von Inhalten.
- **WebSphere MQ Messaging** ermöglicht die Konnektivität verschiedener Systeme, vom Desktop bis zum Großrechner, wobei mehr als 35 Plattformen unterstützt werden; die Kommunikation erfolgt über eine hochqualitative Nachrichtenübertragung.

MQ-Nachrichtenübermittlung

Darunter sind die folgenden Gruppen von **WebSphere MQ** Nachrichtenübermittlungsprodukten zu verstehen:

- **Verteilte Nachrichtenübertragung:** MQ for Windows NT und Windows 2000, AIX, iSeries, HP-UX, Solaris sowie andere Plattformen.
- **Hostnachrichtenübertragung:** MQ for z/OS
- **Universelle (pervasive) Nachrichtenübertragung:** MQe

MQe Darunter ist **WebSphere MQ Everyplace** zu verstehen, die MQ-Produktgruppe zur universellen (pervasiven) Nachrichtenübertragung.

MQI-Kanal

Verbindet einen MQ-Client mit einem WS-Manager auf einem Serversystem und ermöglicht die bidirektionale Übertragung von MQI-Aufrufen und -Antworten.

N

Nachricht

In Anwendungen für die Steuerung von Nachrichtenwarteschlangen der Datenaustausch zwischen Programmen.

Nachrichtenwarteschlange

Siehe *Warteschlange*.

O

Öffentlich

Eine öffentliche Klasse bzw. Schnittstelle ist allgemein sichtbar. Eine öffentliche Methode bzw. Variable wiederum ist überall dort sichtbar, wo auch die zugehörige Klasse sichtbar ist.

Objekt

(1) In Java ist ein Objekt eine Instanz einer Klasse. Eine Klasse modelliert eine Gruppe von Elementen, ein Objekt ein bestimmtes Element dieser Gruppe. (2) In MQ ist ein Objekt ein Warteschlangenmanager, eine Warteschlange oder ein Kanal.

P

Paket Ein Paket ist in Java eine Möglichkeit, einem Java-Programmteil Zugriff auf bestimmte Klassen zu gewähren. Java-Code, der Teil eines bestimmten Paketes ist, hat Zugriff auf alle Klassen in dem Paket und auf alle nicht-privaten Methoden und Felder in den Klassen.

Personal Digital Assistant (PDA)

Ein Taschen-PC.

Privat Ein privates Feld ist nur innerhalb der zugehörigen Klasse sichtbar.

R

Registry

Enthält die Konfigurationsdaten für den WS-Manager.

S

Schnittstelle

Eine Klasse, die nur abstrakte Methoden, jedoch keine Exemplarvariablen enthält. Eine Schnittstelle stellt eine einheitliche Gruppe von Methoden zur Verfügung, die in Form von Unterklassen verschiedener Klassen implementiert werden können.

Server

1. Ein MQe-Server ist ein Gerät, auf dem eine MQe-Empfangsfunktion konfiguriert ist und das in einer Client-Server-Konstellation auf Informationsanforderungen antwortet.
2. Ein MQ-Server ist ein Warteschlangenmanager, der Client-Anwendungen auf einer fernen Workstation Nachrichtenwarteschlangen-Services anbietet.

3. Allgemeiner ist ein Server ein Programm, das in dem speziellen, von zwei Programmen ausgehenden Informationsfluss-Modell von Client und Server auf Informationsanforderungen antwortet.
4. Der Rechner, auf dem ein Server-Programm ausgeführt wird.

Server-Warteschlangenmanager

Ein Warteschlangenmanager mit einer Empfangsfunktion, der folglich Nachrichten nicht nur senden, sondern auch empfangen kann. Siehe MQe-Warteschlangenmanager.

Server-Warteschlangenmanager

Bei MQe: Ein Warteschlangenmanager, zu dem eine Empfangsfunktion hinzugefügt werden kann. Mit der Empfangsfunktion kann er auch Nachrichten empfangen, nicht nur senden.

Serverplattform

Ein Rechner beliebiger Größe, auf dem MQe als Server oder als Client eingesetzt werden kann.

Servlet

Ein Java-Programm, das nur für die Ausführung auf einem Web-Server bestimmt ist.

Superklasse

Eine Klasse, die durch eine anderen Klasse erweitert wurde. Alle öffentlichen und geschützten Methoden bzw. Variablen der Superklasse stehen auch der Unterklasse zur Verfügung.

Synchrone Nachrichtenübertragung

Ein Verfahren zur Kommunikation zwischen Programmen, bei dem Nachrichten in Nachrichtenwarteschlangen gestellt werden. Bei der synchronen Nachrichtenübertragung setzt das Programm, von dem eine Nachricht abgesetzt wurde, die Verarbeitung erst nach Empfang einer Antwort fort. Vgl. *Asynchrone Nachrichtenübertragung*.

T**Transmission Control Protocol/Internet Protocol (TCP/IP)**

Eine Gruppe von Übertragungsprotokollen, die Peer-to-Peer-Konnektivitätsfunktionen für lokale und Weitverkehrsnetze (LANs und WANs) unterstützen.

Transformator

Code, mit dem Daten oder Nachrichten neu formatiert werden.

U**Unterklasse**

Eine Klasse, die zur Erweiterung einer anderen Klasse dient. Die Unterklasse übernimmt alle öffentlichen und geschützten Methoden und Variablen der zugehörigen Superklasse.

V**Verbindung**

Verbindet MQe-Geräte miteinander und ermöglicht die bidirektionale Übertragung synchroner und asynchroner Nachrichten bzw. Antworten.

Verschlüsselungsfunktion

Ein Programm, das Nachrichten verschlüsselt und damit eine sichere Übertragung gewährleistet.

W**Warteschlange**

Eine Warteschlange ist ein MQ-Objekt. Anwendungen für die Steuerung von Warteschlangen können Nachrichten in Warteschlangen einreihen und aus diesen abrufen.

Web Informationen finden Sie im World Wide Web.

Web-Browser

Ein Programm zur Formatierung und Anzeige von Informationen im World Wide Web.

World Wide Web (Web)

Ein Internet-Service, der auf einer einheitlichen Protokollgruppe basiert und bestimmten, hierfür konfigurierten Servern die einheitliche Verteilung von Dokumenten im Internet ermöglicht.

WS-Manager

Ein Systemprogramm, das den Anwendungen Services für die Steuerung von Nachrichtenwarteschlangen zur Verfügung stellt.

Index

A

Adapter, MQe 20
Anwendungen, laden 25
Anwendungen, MQe 9
API 1, 3
Attributregeln 23
Ausgangsserver-Warteschlange 12
Automatische Registrierung 22

B

Begriffe 36
Beispielkonfigurationen 17
Brücke, MQ 31

C

Clientkanäle 9

E

Einführung in MQe 7
Empfangsfunktion 19
Ereignisprotokoll 20

F

Ferne Warteschlange 12

G

Geräte, MQe 9
Gesicherte Nachrichtenzustellung 33
Glossar 36

H

Hinweise, rechtliche 34

K

Kanal 17
Kanäle, Client 9
Klassen, MQe 24
Kompatibilität mit MQ 32
Komprimierung 21
Konfiguration 23
Konfigurationen, Beispiele 17
Kurzeinführung in MQe 8

L

Laden, Anwendungen 25
Lokale Warteschlange 12

M

Marken 35
Minizertifikate, Service für die Vergabe 23
MQ, Kompatibilität mit 32
MQ, Schnittstelle zu 31
MQ-Brücke 31
MQ-Brückenregeln 23
MQ-Brückenwarteschlange 12
MQ-Brückenwarteschlangen 13
MQ-Familie 7
MQ-Nachrichtenaustausch 7
MQ-PCF-Nachrichten 26
MQe 10
MQe-Adapter 20
MQe-Anwendungen 9
MQe-Geräte 9
MQe-Klassen 24
MQe-Regeln 23
MQe-Registry 9, 21
MQe-Sicherheit 21
MQe-Verwaltung 15
MQe-Warteschlangen 11
MQe-Warteschlangenmanager 13

N

Nachrichten 10
Nachrichten, Verwaltung 16
Nachrichtenkonvertierung 31
Nachrichtenzustellung, gesichert 33

O

Öffentliche Registry 22

P

Private Registry 21
Programmierschnittstellen 1, 3

R

rechtliche Hinweise 34
Regeln, benutzerdefinierte Konfiguration 23
Regeln, MQe 23
Registry 21
Registry, MQe 9
Registry, öffentlich 22
Registry, privat 21
Replikation von Zertifikaten 22

S

Schnittstelle, Sicherheit 23
Schnittstelle zu MQ 31

Schnittstellen, Programmierung 1, 3
Sicherheit, MQe 21
Sicherheitsschnittstelle 23

T

Tracefunktion bei MQe 20
Transformatoren 31

U

Übersicht 7
Übertragung 19
Überwachung 16

V

Verbindungen 9, 17
Verbindungen, dynamisch 9
Verbindungsarten 19
Verbindungsmanager 19
Vergabeservice für Minizertifikate 23
Verschlüsselung 21
Verwaltung bei MQe 15
Verwaltung von Einwählverbindungen 20
Verwaltungsnachrichten 16
Vorgänge, WS-Manager 15

W

Warteschlange zum Speichern und Weiterleiten 12
Warteschlangen, ferne 12
Warteschlangen, lokale 12
Warteschlangen, MQ-Brücke 12, 13
Warteschlangen, MQe 11
Warteschlangen, Speichern und Weiterleiten 12
Warteschlangenmanager, MQe 13
Warteschlangenregeln 24
Willkommen 1
WS-Manager 13, 19
WS-Managerregeln 24
WS-Managervorgänge 15

Z

Zertifikatreplikation 22