

MQSeries Everyplace for Multiplatforms



Einführung

Version 1.2

MQSeries Everyplace for Multiplatforms



Einführung

Version 1.2

Anmerkung

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen in „Anhang. Bemerkungen“ auf Seite 91 gelesen werden.

- Die IBM Homepage finden Sie im Internet unter: **ibm.com**
- IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation.
- Das e-business Symbol ist eine Marke der International Business Machines Corporation
- Infoprint ist eine eingetragene Marke der IBM.
- ActionMedia, LANDesk, MMX, Pentium und ProShare sind Marken der Intel Corporation in den USA und/oder anderen Ländern.
- C-bus ist eine Marke der Corollary, Inc. in den USA und/oder anderen Ländern.
- Java und alle Java-basierenden Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- Microsoft Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.
- PC Direct ist eine Marke der Ziff Communications Company in den USA und/oder anderen Ländern.
- SET und das SET-Logo sind Marken der SET Secure Electronic Transaction LLC.
- UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.
- Marken anderer Unternehmen/Hersteller werden anerkannt.

Änderungen in der IBM Terminologie

Die ständige Weiterentwicklung der deutschen Sprache nimmt auch Einfluß auf die IBM Terminologie. Durch die daraus resultierende Umstellung der IBM Terminologie, kann es u. U. vorkommen, dass in diesem Handbuch sowohl alte als auch neue Termini gleichbedeutend verwendet werden. Dies ist der Fall, wenn auf ältere existierende Handbuchausschnitte und/oder Programmteile zurückgegriffen wird.

Dritte Ausgabe (Juni 2001)

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
MQSeries Everyplace for Multiplatforms Introduction Version 1.2,
IBM Form GC34-5843-02,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2000, 2001
© Copyright IBM Deutschland GmbH 2000, 2001

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
Juni 2001

Lizenzhinweise

Bei MQSeries Everyplace Version 1.2 handelt es sich um ein Toolkit, das Benutzern die Erstellung von MQSeries Everyplace-Anwendungen sowie einer Umgebung für deren Ausführung ermöglicht.

Stellen Sie vor Einsatz dieses Produkts oder der Anwendungen, die dieses Produkt verwenden, in einer Produktionsumgebung sicher, dass Sie über die erforderlichen Lizenzen verfügen.

Wenn Sie MQSeries Everyplace auf speziellen Serverplattformen (außer für Codeentwicklung und Tests) verwenden wollen, müssen Sie die *Berechtigungen zur Verwendung in Kapazitätseinheiten* (diese sind in den Berechtigungsnachweisen festgehalten und unterstützen die Verwendung von MQSeries Everyplace gemäß der angegebenen Kapazitätseinheit und Preisgruppentabellen) erwerben, um über die Lizenz zu verfügen, das Programm auf allen Maschinen und Maschinenausrüstungen zu verwenden.

Die *Berechtigungen zur Verwendung auf Einheitenplattformen* (diese sind in den Berechtigungsnachweisen festgehalten und unterstützen die Verwendung von MQSeries Everyplace) sind für die Verwendung des Produkts (außer für Codeentwicklung und Tests) auf den angegebenen Client-Plattformen erforderlich. Mit diesen Lizenzen ist der Benutzer jedoch nicht dazu berechtigt, die Brückenfunktion von MQSeries Everyplace (MQSeries Everyplace Bridge) zu verwenden, oder das Produkt auf den Serverplattformen einzusetzen, die in den Preisgruppenlisten von MQSeries Everyplace aufgenommen sind. Diese Listen wurden von IBM veröffentlicht und stehen auch im Internet unter der nachstehenden URL-Adresse zur Verfügung:

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

Inhaltsverzeichnis

Zu diesem Handbuch	vii	Lokale MQSeries Everyplace-Sicherheit . . .	53
Zielgruppe	vii	MQSeries Everyplace-Sicherheit auf Warte-	
Vorkenntnisse	viii	schlangenebene	54
In diesem Handbuch verwendete Begriffe	viii	Sicherheit auf Nachrichtenebene	55
Kapitel 1. Übersicht	1	Registrierungsdatenbank	57
Kapitel 2. Softwareumgebungen	3	Authentifizierbare MQSeries Everyplace-	
Unterstützte Plattformen	3	Definitionseinheiten	57
Java-Umgebung	4	Private Registrierungsdatenbank und	
Speicherbedarf	5	Berechtigungen	58
Kapitel 3. Die MQSeries-Produktfamilie	7	Automatische Registrierung	59
MQSeries-Produkte für Host-Konfigurationen		Öffentliche Registrierungsdatenbank und	
und verteilte Konfigurationen	9	Zertifikatreplikation	59
MQSeries Everyplace	10	Verwendung der Datenbankservices durch	
Kapitel 4. Produktvoraussetzungen	13	Anwendungen	60
Leistungsspektrum	13	Standardservices für die Vergabe von Mini	
Anwendungen	14	Certificates	60
Kundenspezifische Anforderungen	14	Sicherheitsschnittstelle	61
Kapitel 5. Produktkonzepte	17	Anpassung	61
Einführung	17	Regeln	61
Nachrichtenobjekte	18	Verbindungsarten	63
Format der Ausgabedaten	24	Peer-zu-Peer-Verbindung	63
Warteschlangen	25	Client/Server-Verbindung	64
WS-Manager	32	Mehrere Verbindungsarten	64
WS-Managerkonfiguration	37	Klassen	65
WS-Managervorgänge	40	Anwendungen laden	65
Verbindungen	42	Kapitel 6. MQSeries Everyplace- und	
Verwaltung	44	MQSeries-Netze	67
Verwaltungsnachrichten	44	Schnittstellen zu MQSeries	68
Selektive Verwaltung	46	Nachrichtenumsetzung	76
Überwachung und zugehörige Vorgänge	47	Funktion	78
Dynamische Kanäle	47	Kompatibilität	78
Adapter	49	Gesicherte Zustellung	80
Verwaltung von Einwählverbindungen	49	Kapitel 7. Programmierschnittstellen	81
Trace	50	Kapitel 8. MQSeries Everyplace - Erste	
Ereignisprotokoll	50	Schritte	83
Nachrichtenzustellung	50	MQSeries Everyplace verwenden	84
Asynchrone Nachrichtenzustellung	50	Einstieg	85
Synchrone Nachrichtenzustellung	51	Erster Einsatz von ES02: MQe_Explorer	86
Sicherheit	52	Anhang. Bemerkungen.	91
		Marken	92

Glossar	93	Index	99
Literaturverzeichnis	97	Kommentare an IBM senden	101

Zu diesem Handbuch

Das vorliegende Handbuch enthält eine allgemeine Einführung in das MQSeries Everyplace for Multiplatforms-Produkt, das im Folgenden als MQSeries Everyplace bezeichnet wird. Themen sind die diesem Produkt zu Grunde liegenden Konzepte und das Zusammenspiel mit anderen MQSeries-Produkten.

Ausführliche Informationen zur Anwendungsprogrammierschnittstelle von MQSeries Everyplace und deren Verwendung für die Erstellung von MQSeries Everyplace-Anwendungen finden Sie in den Handbüchern *MQSeries Everyplace for Multiplatforms Programming Reference* und *MQSeries Everyplace for Multiplatforms Programming Guide*.

Weitere Informationen zur Verwendung anderer Programmiersprachen in MQSeries Everyplace for Multiplatforms finden Sie im Handbuch *MQSeries Everyplace for Multiplatforms Native Client Information*.

Informationen zur Vorgehensweise bei der Installation von MQSeries Everyplace for Multiplatforms finden Sie im Handbuch *MQSeries Everyplace for Multiplatforms Wichtige Hinweise*.

Dieses Dokument wird in regelmäßigen Abständen aktualisiert, d. h., es werden ständig neue Informationen und Verbesserungen hinzugefügt. Die neueste Ausgabe finden Sie auf der Webseite mit der Bibliothek für die MQSeries-Produktfamilie unter <http://www.ibm.com/software/ts/mqseries/library/>.

Zielgruppe

Dieses Handbuch richtet sich an alle Benutzer, für die eine sichere Nachrichtenübertragung auf kleinen, kompakten Einheiten wie Sensoren, Telefonen, PDAs (Personal Digital Assistants) und Laptops wichtig ist, sowie an Benutzer, die ihr MQSeries Everyplace-Nachrichtenübertragungsnetz erweitern möchten.

Vorkenntnisse

Zum Verständnis des vorliegenden Handbuchs sind keine Vorkenntnisse erforderlich, Grundkenntnisse in Bezug auf die sichere Nachrichtenübermittlung sind jedoch von Vorteil.

Informationen hierzu finden Sie in dem folgenden MQSeries-Handbuch:

- *MQSeries An Introduction to Messaging and Queuing*

Dieses Handbuch kann als Softcopy unter "Books" aus der MQSeries-Bibliothek im Internet heruntergeladen werden. Diese finden Sie auf der MQSeries-Website unter <http://www.ibm.com/software/ts/MQSeries/library/>.

In diesem Handbuch verwendete Begriffe

Folgende feststehenden Begriffe werden in diesem Handbuch verwendet:

MQSeries-Produktfamilie

Umfasst die folgenden MQSeries-Produkte:

- **MQSeries Workflow** erleichtert die unternehmensweite Integration durch die Automatisierung von Geschäftsprozessen, an denen Menschen und Anwendungen beteiligt sind.
- **MQSeries Integrator** ist eine leistungsstarke Software für die Nachrichtensteuerung; sie ermöglicht intelligentes, regelbasiertes Nachrichten-Routing in Echtzeit sowie Datenumsetzung und -formatierung.
- **MQSeries-Nachrichtenübertragung (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.

MQSeries-Nachrichtenübertragung

Umfasst die folgenden Produktgruppen für Nachrichtenübertragung (Messaging):

- **Verteilte Nachrichtenübertragung:** MQSeries for Windows NT, AIX, AS/400, HP-UX, Sun Solaris sowie andere Plattformen
- **Host-Nachrichtenübertragung:** MQSeries for OS/390
- **Universelle (Pervasive) Nachrichtenübertragung:** MQSeries Everyplace

MQSeries

Umfasst die folgenden drei MQSeries-Produktgruppen für Nachrichtenübertragung:

- Verteilte Nachrichtenübertragung
- Host-Nachrichtenübertragung
- Workstation-Nachrichtenübertragung

MQSeries Everyplace

Umfasst die dritte MQSeries-Produktgruppe für Nachrichtenübertragung, die universelle (*pervasive*) Nachrichtenübertragung.

Einheitenplattform

Ein kleiner Computer, auf dem MQSeries Everyplace lediglich als Client eingesetzt werden kann.

Serverplattform

Ein Computer beliebiger Größe, auf dem MQSeries Everyplace als Server oder Client eingesetzt werden kann.

Gateway

Ein Computer beliebiger Größe, auf dem MQSeries Everyplace-Programme (einschließlich der MQSeries-Brückenfunktion) ausgeführt werden.

Kapitel 1. Übersicht

MQSeries Everyplace gehört zur MQSeries-Produktfamilie für Nachrichtenübertragung (Business Messaging). Dieses Produkt erfüllt die Anforderungen der Nachrichtenübertragung bei kleinen Einheiten wie Sensoren, Telefonen, Personal Digital Assistants (PDAs) und Laptops und unterstützt darüber hinaus die Mobilität und auch die Anforderungen, die sich aus der Verwendung nicht sehr stabiler DFV-Netze ergeben. Es sorgt dafür, dass Nachrichten nur einmal, doch dafür sicher und zuverlässig übertragen werden (ein Standardmerkmal von MQSeries) und ermöglicht den Nachrichtenaustausch mit anderen MQSeries-Produkten. Da viele MQSeries Everyplace-Anwendungen außerhalb des Schutzes einer Internet-Firewall ausgeführt werden, stellt dieses Produkt darüber hinaus leistungsstarke Sicherheitsfunktionen zur Verfügung.

Kleine Einheiten benötigen ein Messaging-Subsystem, das die Systemressourcen möglichst wenig belastet; daher bietet MQSeries Everyplace Funktionen und Schnittstellen, die auf diesen Benutzerkreis zugeschnitten sind und deren Umfang aus diesem Grund auch nicht dem sonst bei anderen MQSeries-Produkten üblichen Umfang entspricht. Dafür bietet dieses Produkt jedoch eine Reihe von Sonderfunktionen für genau diesen Benutzerkreis; dazu gehören umfassende Sicherheitsvorrichtungen, Nachrichtenobjekte, synchrone und asynchrone Nachrichtenübertragung, Zugriff auf ferne Warteschlangen sowie die Nachrichtenübertragung im Push- und Pull-Modus.

Darüber hinaus arbeitet MQSeries Everyplace problemlos mit anderen Produkten aus der IBM Pervasive Computing-Produktfamilie sowie mit anderen Komponenten von Websphere Application Server.

Kapitel 2. Softwareumgebungen

Unterstützte Plattformen

Eine direkte Installation von MQSeries Everyplace ist nur auf einigen Serverplattformen möglich. Sollen Programme und Java-Klassen auf eine andere Plattform übertragen werden, muss ein entsprechendes Übertragungs- bzw. Dateiübertragungsprogramm verwendet werden, das nicht im Lieferumfang enthalten ist.

Direkt unterstützte Plattformen mit Installationsunterstützung

MQSeries Everyplace kann auf folgenden Plattformen unter Verwendung der integrierten Tools installiert werden:

- Windows NT Version 4
- Windows 2000
- Windows 95/98/ME
- AIX Version 4.3
- Sun Solaris Version 7 oder 8
- Linux Intel Kernel 2.2 (wird über eine Zip-Datei installiert)
- HP-UX 11.0 (wird über eine Zip-Datei installiert)

Direkt unterstützte Plattformen jedoch ohne Installationsunterstützung

Folgende Plattformen werden für den Test und den Einsatz von MQSeries Everyplace unterstützt; es ist jedoch nur eine Installation per Dateiübertragung von einer anderen Plattform möglich.

- WinCE 2.1 auf HP Jornada-Einheiten (Modell 680 und 820)
- EPOC Release 5, das 32-Bit-Betriebssystem, auf PSION-Einheiten (5MX Pro oder NetBook)
- PalmOS, Version 3.0 oder höher auf Palm V und IBM Workpad C3
- IBM 4690 OS mit Java

Softwareumgebungen

Indirekt unterstützte Plattformen

Folgende Plattformen können verwendet werden; sie werden jedoch nur unterstützt, wenn ihre Java-Umgebung vollständig mit der Umgebung auf den direkt unterstützten Plattformen kompatibel ist. Eine Fehlerbestimmung ist nur auf einer der oben aufgeführten getesteten Plattformen möglich.

- Linux auf IBM zSeries mit Kernel 2.2
- iSeries
- OS/2
- EPOC (auf anderen als den oben aufgeführten Einheiten)
- WinCE (auf anderen als den oben aufgeführten Einheiten)
- QNX Neutrino
- Pocket PC
- PalmOS (auf anderen als den oben aufgeführten Einheiten)
- Jede Plattform mit einer der unter „Java-Umgebung“ aufgeführten Java-Umgebungen

Java-Umgebung

Eine der folgenden Java-Laufzeitumgebungen ist erforderlich:

- IBM Java Runtime (JVM 1.3 oder höher) einschließlich Java Micro Edition
- Alle Java-Umgebungen, die Sun-zertifiziert sind (Version 1.1 oder höher)¹

Anmerkung: Für einen problemlosen Betrieb wird vorausgesetzt, dass die Java-Umgebung vollständig mit der auf einer der oben angeführten Plattformen getesteten Java-Umgebung kompatibel ist.

- HP Jornada-Einheiten (Modell 680 und 820) mit Windows CE
- PSION-Einheiten (5MX Pro oder NetBook) mit EPOC
- Eine der oben aufgeführten direkt unterstützten Serverplattformen.

Für die MQSeries-Brückenfunktion ist MQSeries Classes for Java erforderlich. Überprüfen Sie, welche Java-Version erforderlich ist, um die betreffende Version von MQSeries Classes for Java auszuführen.

1. Bei Ausführung des Installationsprogramms unter der JVM von SUN treten unter Umständen Probleme auf, wenn der JIT-Compiler (Just In Time) aktiviert ist. Sie können diesen Compiler mit dem Befehl `java -Djava.compiler=NONE install` inaktivieren.

Speicherbedarf

In der folgenden Tabelle ist der jeweilige Speicherbedarf aufgeführt, der für die Installation von MQSeries Everyplace erforderlich ist.

Tabelle 1. Erforderlicher Speicherplatz für die Installation

Betriebssystem	Erforderlicher Speicherplatz
Windows NT (Dateisystem = NTFS)	26 MB
AIX	29 MB
Solaris	27 MB

In der folgenden Tabelle ist der jeweilige Speicherbedarf aufgeführt, der nach der Installation von MQSeries Everyplace benötigt wird.

Tabelle 2. Erforderlicher Speicherplatz für MQSeries Everyplace

Betriebssystem	Erforderlicher Speicherplatz
Windows NT (Dateisystem = NTFS)	9,5 MB
AIX	11 MB
Solaris	10 MB

Kapitel 3. Die MQSeries-Produktfamilie

Zur MQSeries-Produktfamilie gehört eine Vielzahl an Produkten, die die in Abb. 1 dargestellten Merkmale und Funktionen zur Verfügung stellen.

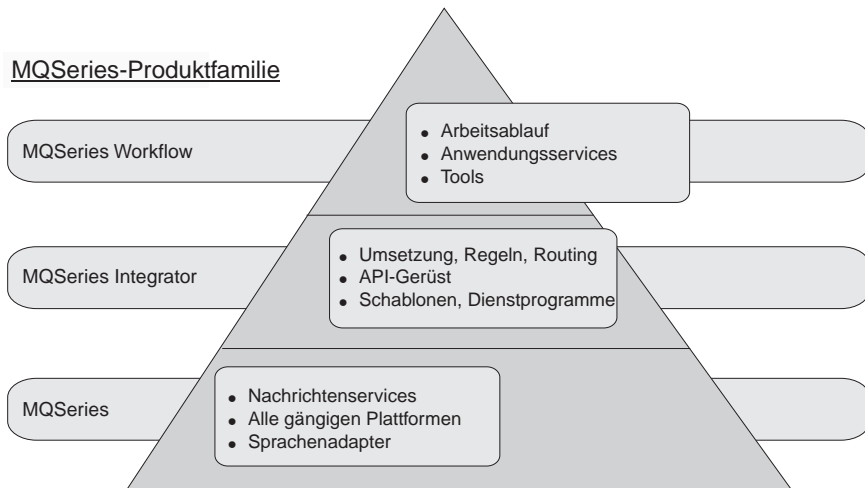


Abbildung 1. MQSeries-Produktfamilie

- **MQSeries Workflow** erleichtert die unternehmensweite Integration durch die Automatisierung von Geschäftsprozessen, an denen Menschen und Anwendungen beteiligt sind.
- **MQSeries Integrator** ist eine leistungsstarke Software für die Nachrichtensteuerung; sie ermöglicht intelligentes, regelbasiertes Nachrichten-Routing in Echtzeit sowie Datenumsetzung und -formatierung.
- **MQSeries-Nachrichtenübertragung (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.

MQSeries Workflow und MQSeries Integrator bieten Konnektivität über die MQSeries-Nachrichtenübertragungsschicht.

MQSeries-Produktfamilie

Die MQSeries-Nachrichtenübertragung wird sowohl von MQSeries- als auch von MQSeries Everyplace-Produkten zur Verfügung gestellt; alle Produkte unterstützen ein oder mehrere Hardwareserverplattformen bzw. die entsprechenden Betriebssysteme. Da die einzelnen Plattformen ein unterschiedliches Leistungsspektrum aufweisen, wurden die einzelnen Produkte in Produktgruppen zusammengefasst, denen jeweils ein gemeinsames Konzept zu Grunde liegt und die einen bestimmten Bereich abdecken. Es handelt sich hierbei um drei Produktgruppen:

- **Verteilte Nachrichtenübertragung:** MQSeries for Windows NT, AIX, AS/400, HP-UX, Sun Solaris sowie andere Plattformen
- **Host-Nachrichtenübertragung:** MQSeries for OS/390
- **Universelle (Pervasive) Nachrichtenübertragung:** MQSeries Everyplace

Die Nachrichtenübertragung selbst erfolgt bei allen Produktgruppen bzw. Produkten über Warteschlangenmanager (WS-Manager). Diese verwalten Warteschlangen, in denen Nachrichten gespeichert werden können. Anwendungen kommunizieren jeweils mit einem lokalen WS-Manager, über den sie Nachrichten in Warteschlangen stellen bzw. aus diesen abrufen. Nachrichten, die in eine ferne, einem fernen WS-Manager zugeordnete Warteschlange eingereicht werden, werden über Kanäle 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. Grundlegendes Konzept der Nachrichtenübertragung ist eine Aufhebung der Abhängigkeit zwischen sendender und empfangender Anwendung, indem Nachrichten an zwischengeschalteten Punkten bei Bedarf in Warteschlangen eingereicht werden. Alle MQSeries-Produkte, die auf dieser Nachrichtenübertragung beruhen, arbeiten mit denselben grundlegenden Komponenten, nämlich WS-Managern, Warteschlangen, Nachrichten und Kanälen, wobei jedoch im Detail zahlreiche Unterschiede zwischen den einzelnen Produkten bestehen.

MQSeries-Produkte für Host-Konfigurationen und verteilte Konfigurationen

MQSeries-Produkte für die Nachrichtenübertragung in Host-Umgebungen und verteilten Umgebungen unterstützen verschiedene Netzkonfigurationen mit Servern und Clients.² Einige Beispielkonfigurationen sind in Abb. 2 zu sehen.

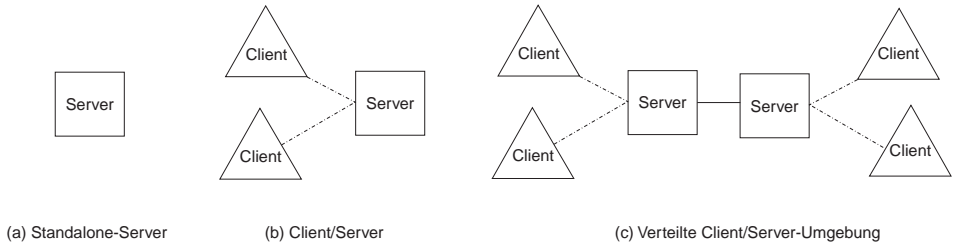


Abbildung 2. Einfache Host-Konfiguration und verteilte Konfigurationen

Bei der einfachsten Konfiguration handelt es sich um einen Standalone-Server mit einem WS-Manager. Auf diesem Server kommen eine oder mehrere Anwendungen zur Ausführung, die Nachrichten über Warteschlangen miteinander austauschen. Eine Alternative ist die Client/Server-Konfiguration. In dieser Konfiguration befindet sich der WS-Manager nur auf dem Server; die einzelnen Clients greifen auf diesen WS-Manager über einen *Client-Kanal* zu. Bei diesem Client-Kanal handelt es sich um eine bidirektionale Übertragungsverbindung mit einem spezifischen MQSeries-Protokoll, das auf einem Aufrufmechanismus ähnlich dem RPC-Verfahren (Remote Procedure Call) basiert. Die Anwendungen können auf dem Client ausgeführt werden und verfügen über Zugriff auf die Serverwarteschlangen. Ein Vorteil der Client/Server-Konfiguration ist, dass die Client-Nachrichtenübertragung wenig Ressourcen belegt, da sich der WS-Manager auf dem Server befindet. Nachteilig an dieser Konfiguration ist die Tatsache, dass die Clients und der zugehörige Server synchron arbeiten, d. h., die Client-Kanäle müssen jederzeit verfügbar sein.

Ein komplexeres Szenario ist die verteilte Client/Server-Konfiguration mit mehreren Servern. In dieser Konfiguration tauschen die Server Nachrichten über *Nachrichtenkanäle* aus. Diese Nachrichtenkanäle sind unidirektional und arbeiten mit einem Protokoll, das für den sicheren, asynchronen Austausch von Nachrichtendaten konzipiert ist. Die Clients können die Verarbeitung fortsetzen, auch wenn diese Nachrichtenkanäle nicht verfügbar sind; allerdings ist in diesem Fall kein Nachrichtenaustausch zwischen den Servern möglich.

2. Diese Begriffe haben eine bestimmte Bedeutung in den MQSeries-Messaging-Produkten für Host- und Workstation-Umgebungen und verteilte Umgebungen, die nicht immer der sonst üblichen Bedeutung entspricht.

MQSeries Everyplace

Über WS-Manager unterstützt MQSeries Everyplace eine Vielzahl von Netzkonfigurationen, wobei jeder WS-Manager über entsprechende Merkmale verfügt. Das Client- oder Serverkonzept wie bei MQSeries-Produkten für Host-Umgebungen oder verteilte Umgebungen gilt hier nicht. Die WS-Manager in MQSeries Everyplace können zwar die Rolle traditioneller Clients oder Server übernehmen, tatsächlich handelt es sich jedoch lediglich um WS-Manager, die für die Ausführung anwendungsspezifischer Aufgaben konfiguriert sind. So kann ein WS-Manager in MQSeries Everyplace beispielsweise mit oder ohne lokale Warteschlangen konfiguriert werden. Wird er mit lokalen Warteschlangen konfiguriert, kann er Nachrichten lokal speichern und damit Anwendungen die asynchrone Nachrichtenübertragung ermöglichen; ohne lokale Warteschlangen hingegen ist nur eine synchrone Nachrichtenübertragung möglich. Ein weiteres Beispiel für eine solche spezifische Anpassung ist die Konfiguration eines WS-Managers mit oder ohne Brückenfunktionen. Mit der Brückenfunktion kann ein WS-Manager Nachrichten mit MQSeries-WS-Managern auf Hosts oder Workstations austauschen; ohne Brückenfunktion ist nur ein Austausch mit anderen MQSeries Everyplace-WS-Managern möglich (über andere WS-Manager mit Brückenfunktion im Netz ist jedoch zumindest eine indirekte Kommunikation möglich).

Anmerkung: Ein neuer Knoten für MQSeries Integrator (MQSI) ermöglicht jetzt eine direkte Verbindung zu MQSeries Everyplace ohne die MQSeries-Brückenfunktion.

WS-Manager in MQSeries Everyplace tauschen Daten über *dynamische Kanäle* aus. Die Merkmale dieser dynamischen Kanäle unterscheiden sich von denen der *Client-* und *Nachrichtenkanäle*, die von anderen Produkten der Produktfamilie verwendet werden. Hier einige der wichtigsten Merkmale dynamischer Kanäle:

- Unterstützung der *synchronen* und *asynchronen* Nachrichtenübertragung: 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.
- *Endpunkt-zu-Endpunkt-Übertragung*: Die Kanäle führen vom Quellen- zum Ziel-WS-Manager, wobei zwischengeschaltete WS-Manager möglich sind. Das im Kanal verwendete Übertragungsprotokoll kann sich auf dem Weg durch diese zwischengeschalteten WS-Manager ändern.

- Unterstützung von *Komprimierung*, *Verschlüsselung* und *Authentifizierung*: Kanäle bieten diese Sicherheitsmerkmale, um die Daten bei der Übertragung zu schützen.
- Unterstützung des *Peer-zu-Peer*- und des *Client-Server*-Betriebs: Kanäle in einer Peer-zu-Peer-Konfiguration arbeiten symmetrisch, d. h., der Quellen- oder der Ziel-WS-Manager kann den Kanalbetrieb starten. Client-Server-Kanäle dagegen arbeiten nach dem Anforderungs-/Antwort-Prinzip: Der Client sendet eine Anforderung an den Server, der auf diese antwortet. (Dies bedeutet keine Einschränkung für den Nachrichtenfluss. Dieser kann vom Client zum Server und vom Server zum Client erfolgen.)

Die Brückenkonfiguration ermöglicht einem MQSeries Everyplace-WS-Manager über *Client-Kanäle* die Kommunikation mit MQSeries-WS-Managern in Host-Umgebungen und verteilten Umgebungen. Die Brückekomponente verwaltet einen Pool mit Client-Kanälen, die einem oder mehreren WS-Managern in einer Host-Umgebung oder einer verteilten Umgebung zugeordnet werden können. Mehrere MQSeries Everyplace-WS-Manager mit Brückenfunktion können zu einem Netz zusammengeschlossen werden, das die erforderliche Kapazität, Leistung und Zuverlässigkeit bietet.

Die folgenden Abbildungen zeigen einige typische MQSeries Everyplace-Konfigurationen. Aus Gründen der Übersichtlichkeit sind nur die definierten Direktverbindungen zu sehen. Es besteht jedoch auch die Möglichkeit, indirekte Verbindungen über Direktverbindungen zu definieren. Verbindungslinien mit einem Pfeil an jedem Ende stehen für Peer-zu-Peer-Kanäle, Verbindungslinien mit einem Pfeil nur an einem Ende stehen für Client/Server-Kanäle, wobei der Pfeil auf den Server weist. Clients können die Client/Server-Kanäle für die Nachrichtenübertragung an den Server und für den Nachrichtenempfang vom Server im Pull-Modus verwenden. Verbindungslinien ohne Pfeil stehen für MQSeries-Client-Kanäle, die eine Kommunikation zwischen MQSeries Everyplace und MQSeries ermöglichen.



(a) Standalone-WS-Manager

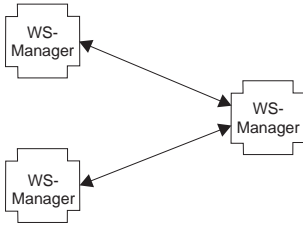
(a) Ein Standalone-WS-Manager, der eine oder mehrere Anwendungen unterstützt, die für den Datenaustausch Warteschlangen verwenden.



(b) Peer-zu-Peer-Konfiguration

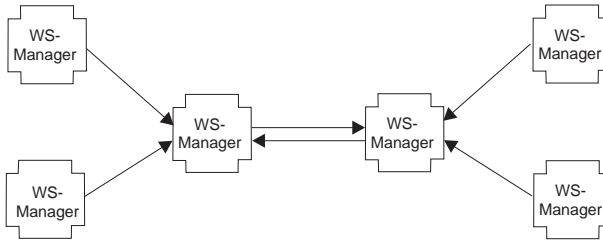
MQSeries-Produktfamilie

(b) Zwei WS-Manager, die über einen Peer-zu-Peer-Kanal miteinander verbunden sind.



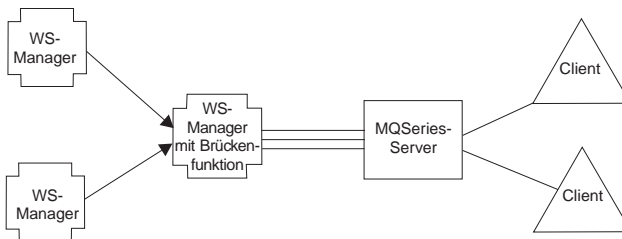
(c) Peer-zu-Peer-Konfiguration

(c) Hier ist ein weiterer WS-Manager über einen Peer-zu-Peer-Kanal mit einem dritten WS-Manager verbunden. Alle drei WS-Manager können über die direkten Verbindungen Daten miteinander austauschen.



(d) Kleines Netz

(d) Eine kleine Netzkonfiguration; die zentralen Server-WS-Manager tauschen über zwei direkte Client/Server-Kanäle Daten aus. Die Client-WS-Manager sind jeweils über einen direkten Client/Server-Kanal mit einem der Server-WS-Manager verbunden.



(e) Netz mit integrierter MQSeries-Produktfamilie

(e) Eine MQSeries Everyplace-Konfiguration, in der einer der WS-Manager über die Brückenfunktion verfügt; der Pool der Client-Kanäle ist mit einem einzelnen MQSeries-Zielservers in einer Host- oder in einer verteilten Umgebung zugeordnet.

Kapitel 4. Produktvoraussetzungen

Dieses Kapitel setzt sich mit den Anforderungen auseinander, die bei der Entwicklung und Implementierung von MQSeries Everyplace berücksichtigt wurden.

Leistungsspektrum

MQSeries Everyplace erweitert die Nachrichtenübertragungsmöglichkeiten der MQSeries-Produktfamilie wie folgt:

- Unterstützt kleine Einheiten wie PDAs, Telefone und Sensoren, die damit in ein MQSeries-Netz eingebunden werden können. Darüber hinaus unterstützt MQSeries Everyplace auch zwischengeschaltete Einheiten wie Laptops und Workstations sowie verteilte Plattformen und Host-Plattformen. MQSeries Everyplace bietet dieselbe Betriebsgüte, die gesicherte einmalige Nachrichtenzustellung, und ermöglicht den Nachrichtenaustausch mit anderen Produkten der Produktfamilie.
- Stellt Messaging-Funktionen zur Verfügung, die wenig Ressourcen belegen.
- Stellt umfassende Sicherheitseinrichtungen zum Schutz von Warteschlangen sowie von Nachrichten und zugehörigen Daten (während diese gespeichert sind bzw. übertragen werden) zur Verfügung.
- Ermöglicht einen effizienten Betrieb in unzuverlässigen Übertragungs-umgebungen mit instabilen Netzen oder stark eingeschränkter Bandbreite. MQSeries Everyplace verfügt über ein effizientes Übertragungsprotokoll und ermöglicht die automatische Wiederherstellung bei Ausfall einer DFV-Verbindung.
- Unterstützt mobile Benutzer, indem es beim Roaming den Wechsel der Netzanschlusspunkte ermöglicht. Darüber hinaus ermöglicht MQSeries Everyplace auch eine Steuerung des Verhaltens bei schwachen bzw. leeren Batterien und Netzeinschränkungen bzw. Netzausfällen.
- Ermöglicht den Einsatz entsprechend konfigurierter Firewalls.
- Erfordert nur einen minimalen Verwaltungsaufwand vom Benutzer; dadurch ist die Installation von MQSeries Everyplace nahezu transparent. MQSeries Everyplace stellt daher eine geeignete Basis für die Erstellung aufgabenspezifischer Anwendungen dar.
- Ermöglicht problemlose Anpassungen und Erweiterungen über anwendungsspezifische Regeln und andere Klassen, über die das Verhalten entsprechend geändert werden kann, oder über die Erweiterung von Basisobjektklassen durch Unterklassen, um beispielsweise verschiedene Nachrichtenarten zu realisieren.

Anwendungen

Es gibt eine Vielzahl möglicher MQSeries Everyplace-Anwendungen, doch wird davon ausgegangen, dass es sich bei einer großen Zahl der Anwendungen um benutzerdefinierte Anwendungen handelt, die für bestimmte Benutzergruppen entwickelt werden. Hier einige Beispiele:

- **Handelsanwendungen:** Übertragung einzelner Kassentransaktionen an Host-Systeme wie beispielsweise Nachrichten-Broker.
- **Kundenanwendungen:** Lebensmitteleinkäufe von zu Hause über einen PDA, Ermittlung von Kundenwünschen bei Fluglinien, finanzielle Transaktionen über mobile Telefone.
- **Steueranwendungen:** 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.
- **Mobiler Arbeitseinsatz:** Vertreter (z. B. Versicherungsagenten); schnelle Übermittlung von Empfangsbestätigungen für Paketzustellungsdienste; Kommunikation in einem Schnellimbiss zwischen Angestellten und Küche; Punktezahlung bei Golfturnieren; Nachrichtensysteme für die bei der Polizei eingesetzten mobilen gesicherten Systeme; fachspezifische Informationen für Angestellte bei Versorgungseinrichtungen in Situationen, in denen es zu häufigen Verbindungsabbrüchen kommt, Zählerablesung in Privathaushalten.
- **Anwendungen für den individuellen Einsatz:** Post-/Kalenderreplikation, Datenbankreplikation, beansprucht wenig Platz auf Laptops.

Kundenspezifische Anforderungen

Folgende Anforderungen wurden u. a. bei der Konzipierung von MQSeries Everyplace berücksichtigt:

- **Verwaltung:** Minimaler Installations- und Wartungsaufwand; Unterstützung der lokalen und fernen Verwaltung; anwendungsspezifische Erweiterungs- und Anpassungsmöglichkeiten der Verwaltungsfunktionen; automatische Fehlererkennung und -behebung; unabhängige Verwaltungskomponenten, die gezielt eingesetzt werden können.
- **Kommunikation:** Leistungsstarkes Übertragungsprotokoll; minimale Header; keine obligatorischen Felder (mit Ausnahme einer eindeutigen ID); Möglichkeit zum Ändern der Datencodierung; Unterstützung von Komprimierung, Verschlüsselung und Authentifizierung; Endpunkt-zu-Endpunkt-Festlegung der Komprimierung und der Sicherheitsmerkmale; Kommunikation über eine Firewall hinweg; steckbare Kommunikationsadapter.

- **Kompatibilität:** MQSeries-Qualität und problemloser Nachrichtenaustausch mit anderen MQSeries-Produkten; Kommunikation mit vorhandenen MQSeries-Systemen, ohne dass Änderungen an Anwendungen erforderlich sind; flexible Steuerung des Nachrichtenaustausches zwischen MQSeries und MQSeries Everyplace.
- **Speicheranforderung:** 64 KB für Palm-Einheiten. Für Java-Einheiten sind mindestens 100 KB für die Java-Klassendateien erforderlich.
- **Funktionen:** Synchrone und asynchrone Nachrichtenübertragung; Zugriff auf Nachrichten in lokalen oder fernen Warteschlangen; gezielter Abruf einzelner Nachrichtenfelder; gezielte Steuerung des Backup-Datenträgers einer Warteschlange.
- **Regelunterstützung:** Steuerung des Verhaltens über Regeln, z. B. Zeitpunkt der Nachrichtenübertragung, Anzahl der Verbindungsversuche, Handhabung von zu langen Nachrichten, Maßnahmen, wenn eine Zielwarteschlange voll ist.
- **Sicherheit:** Vollständige Unterstützung von Sicherheitseinrichtungen, Authentifizierung und digitalen Unterschriften (Non-Repudiation); Sicherheit auf Nachrichten- und Warteschlangenebene; Schutz des Nachrichtenübertragungssystems vor unberechtigten Zugriffen; "steckbare" Sicherheitseinrichtungen über Algorithmen, die dem Industriestandard entsprechen; Möglichkeit zur Integration der betriebssystemspezifischen Benutzerberechtigungen; kann nationalen Sicherheitsbestimmungen entsprechend konfiguriert werden, wobei sich die Sicherheitsunterstützung bei der grenzüberschreitenden Nachrichtenübertragung jeweils ändert.

Produktvoraussetzungen

Kapitel 5. Produktkonzepte

Einführung

Die grundlegenden Komponenten des MQSeries Everyplace-Programmiermodells sind Nachrichten, Warteschlangen und WS-Manager. Bei MQSeries Everyplace-Nachrichten handelt es sich um Objekte mit anwendungsspezifischem Inhalt. Nachrichten werden in Warteschlangen gespeichert und können auf diese Weise über ein MQSeries Everyplace-Netz übertragen 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. Konfigurationsdaten werden in einer Registrierdatenbank gespeichert.

Das folgende Fenster des Verwaltungs-Tools MQe_Explorer enthält eine Übersicht über die MQSeries Everyplace-Objektstruktur:

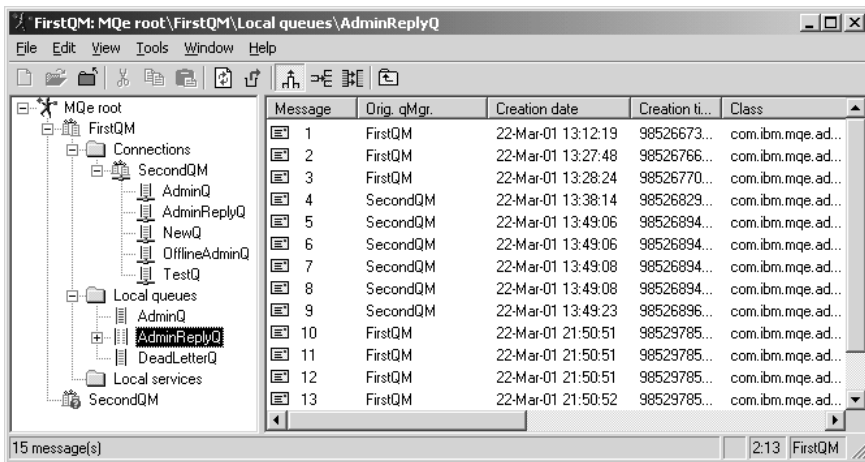


Abbildung 3. MQSeries Everyplace-Objekte im MQe_Explorer

Im linken Teilfenster sind die Objekte in Form einer Baumstruktur angeordnet. Unterhalb des obersten Elements in der Baumstruktur, MQe root, sehen Sie die beiden WS-Manager FirstQM und SecondQM. Die Struktur unterhalb von SecondQM wurde erweitert, so dass die Ordner *Connections* (Verbindungen), *Local queues* (Lokale Warteschlangen) und *Local services* (Lokale Services) zu sehen sind. Bei einer Erweiterung des Ordners *Connections* ist eine

Produktkonzepte

Verbindungsdefinition von WS-Manager SecondQM zu WS-Manager FirstQM zu sehen. Der ferne WS-Manager SecondQM verfügt über die Definitionen von fünf Warteschlangen des WS-Managers FirstQM. Bei einer Erweiterung des Ordners *Local queues* unterhalb von SecondQM sind drei lokale Warteschlangen zu sehen; eine dieser Warteschlangen ist ausgewählt, ihr Inhalt ist im rechten Teilfenster zu sehen.

Ein Verständnis von MQSeries Everyplace setzt Kenntnisse der Konzepte von Nachrichten, Warteschlangen, WS-Managern und Verbindungen voraus. Eine Übersicht über diese Konzepte finden Sie in den folgenden Abschnitten.

Nachrichtenobjekte

Es bestehen grundsätzliche Unterschiede zwischen MQSeries Everyplace-Nachrichtenobjekten und den von der MQSeries-Nachrichtenübermittlung unterstützten Nachrichten. In MQSeries handelt es sich bei Nachrichten um Bytebereiche, die sich aus Nachrichten-Header und dem Nachrichtenhauptteil zusammensetzen. Der Nachrichten-Header ist lesbar und enthält wichtige Informationen, wie beispielsweise den Namen der Warteschlangen für Antwortnachrichten und den Namen des WS-Managers, an den die Nachrichten gerichtet sind, die Nachrichten-ID und die Korrelations-ID. Der Nachrichtenhauptteil ist für das Produkt nicht lesbar.

In MQSeries Everyplace hingegen handelt es sich bei Nachrichten um *Nachrichtenobjekte*, die von einem MQSeries Everyplace-Objekt, dem so genannten *Feldobjekt*, übernommen werden. Diese Nachrichten haben keinen Header oder Nachrichtenhauptteil, sie verfügen jedoch über Merkmale und Methoden. Ein Verständnis dieser Nachrichtenobjekte setzt ein Verständnis des übergeordneten Feldobjekts voraus, von dem diese abgeleitet werden.

Feldobjekte sind ein wichtiger Bestandteil in MQSeries Everyplace; es handelt sich dabei um eine Gruppe von *Feldern*, wobei sich jedes Feld aus dem Namen, einem Datentyp und den eigentlichen Daten zusammensetzt. Feldnamen sind ASCII-Zeichenfolgen (in denen die Verwendung einiger reservierter Zeichen nicht zulässig ist), deren Länge nicht begrenzt ist.

Feldobjekten ist eine *Feldart* zugeordnet, bei der es sich um eine abgekürzte Zeichenfolge handelt, die dem Namen der Programmierobjektklasse entspricht.

Folgende Feldarten sind möglich:

Tabelle 3. Feldobjektarten

Art	Beschreibung
ASCII	ASCII-Zeichenfolge oder dynamischer ASCII-Zeichenfolgebereich
Boolescher Wert	Wert
Byte	Fester oder dynamischer Bytewertbereich
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	Wert (4 Bytes) oder fester bzw. dynamischer Integerbereich
Langes Integer	Wert (8 Bytes) oder fester bzw. dynamischer Bereich langer Integer
Objekt	Wert
Kurzes Integer	Wert (2 Bytes) oder fester bzw. dynamischer Bereich kurzer Integer
Unicode	Unicode-Zeichenfolge oder dynamischer Bereich von Unicode-Zeichenfolgen

Feldobjekte unterstützen verschiedene Methodenaufrufe; Beispiel:

- Aufzählung von Feldern
- Kopieren eines oder mehrerer Ursprungsfelder (mit der Ersetzungsoption)
- Vergleich von Feldobjekten
- Schreiben in/Lesen aus Ursprungsfeldern
- Überprüfung von Feldobjekten (ob beispielsweise ein Ursprungsfeld in einem Objekt vorhanden ist)
- Ausgabe und Wiederherstellung in einen Bytebereich bzw. aus einem Bytebereich

Nachrichtenobjekte

Die Ausgabe- und die Wiederherstellungsmethode sind von besonderer Bedeutung. Die Ausgabemethode wird beispielsweise aufgerufen, um Daten für die Übertragung eines Nachrichtenobjekts über eine DFV-Verbindung zur Verfügung zu stellen, während die Wiederherstellungsmethode die Wiederherstellung des Objekts auf der Empfängerseite ermöglicht. Das Objekt ist auf diese Weise für das eigene Übertragungsformat verantwortlich; die Standardmethoden, die zur Verfügung gestellt werden, können durch benutzerdefinierte Methoden ersetzt werden. Ebenso werden diese beiden Methoden verwendet, wenn Nachrichtenobjekte in Warteschlangen ausgegeben bzw. aus Warteschlangen wiederhergestellt werden. Durch Überschreiben dieser Methoden kann das Verhalten grundlegend geändert werden. Beispielsweise kann ein Nachrichtenobjekt bei der Ausgabe eine Datenbank abfragen, um vor der Übertragung den Wert zu extrahieren.

In der folgenden Tabelle sind einige der Merkmale von Feldobjekten und die Ursprungsfelder, die sie enthalten, aufgeführt:

Tabelle 4. Feldobjekte und ihre Feldmerkmale

Merkmal	Vorhandensein	
	Feldobjekte	Felder
Zugeordnetes Attributobjekt	Optional	
Ursprungsfelder	Ja	
Verdeckt		Ja
Name		Ja
Art	Ja	Ja
Wert		Ja

Von besonderer Bedeutung sind zwei Merkmale, nämlich *Verdeckt*, mit dem ein Feld für Vergleichszwecke ignoriert werden kann, und die Möglichkeit, einem Feldobjekt *ein Attributobjekt zuzuordnen*.

Attributobjekte sind von grundlegender Bedeutung für das MQSeries Everyplace-Sicherheitskonzept und ermöglichen den selektiven Zugriff auf den Inhalt sowie den Schutz des Inhalts. Sie verfügen über die folgenden wichtigen Merkmale:

Tabelle 5. Attributobjekt - Merkmale

Merkmal	Beschreibung
Authentifizierung	Ermöglicht die Zugriffssteuerung
Verschlüsselung	Schützt den Objektinhalt bei Ausgabe in einen Bytebereich (und erlaubt die Wiederherstellung dieses Inhalts)

Table 5. Attributobjekt - Merkmale (Forts.)

Merkmals	Beschreibung
Komprimierung	Reduziert den Speicherbedarf (für die Übertragung und/oder die Speicherung)
Regel	Ermöglicht die Steuerung zulässiger Vorgänge

Nachrichtenobjekte übernehmen Merkmale von Feldobjekten und verfügen daher über alle oben aufgeführten Merkmale, einschließlich der Möglichkeit, dass ihnen Attributobjekte auf der Ebene von Nachrichten und Ursprungsfeldobjekten zugeordnet werden können. Darüber hinaus ist Nachrichtenobjekten eine UID (Unique Identifier, eindeutige Kennung) zugewiesen, die von MQSeries Everyplace generiert wird. Jedem Nachrichtenobjekt im MQSeries Everyplace-Netz wird eine UID zugewiesen, die sich aus Folgendem zusammensetzt:

- Dem **Namen** des Ursprungs-WS-Managers (wird vom WS-Manager bei Empfang des Objekts hinzugefügt). Hierbei muss es sich um einen global eindeutigen Namen handeln.
- Dem **Zeitpunkt**, zu dem das Nachrichtenobjekt erstellt wurde (wird bei der Erstellung hinzugefügt).

Daher verfügt ein Nachrichtenobjekt zusätzlich zu den vom übergeordneten Feldobjekt übernommenen Merkmalen noch über folgende Merkmale:

Table 6. Nachrichtenobjekt - Merkmale

Merkmals	Beschreibung
Quellen-WS-Manager	Der Name des WS-Managers, der die neue Nachricht zuerst von der Anwendung erhielt
Erstellungszeit	Der Zeitpunkt, zu dem das Nachrichtenobjekt von der Anwendung erstellt wurde
UID	Eindeutige MQSeries Everyplace-Kennung

Für Nachrichten, die an einen anderen MQSeries Everyplace-WS-Manager gerichtet sind, sind keine weiteren Informationen erforderlich, obwohl in der Regel wahrscheinlich weitere Felder vorhanden sein werden. Bei diesen zusätzlichen Feldern handelt es sich beispielsweise um folgende:

- Felder, die von MQSeries Everyplace hinzugefügt werden, um den aktuellen Status anzuzeigen
- Felder, die einer bestimmten Nachrichtenunterklasse zugeordnet sind
- Angepasste Felder, die einem bestimmten Nachrichtenobjektexemplar zugeordnet sind

Nachrichtenobjekte

MQSeries Everyplace fügt einem Nachrichtenobjekt Felder hinzu bzw. entfernt Felder aus einem Nachrichtenobjekt, um Messaging- und Queuing-Vorgänge auszuführen. Befindet sich eine Nachricht beispielsweise in einer Warteschlange, kann die Nachricht auf den Zeitpunkt hin abgefragt werden, zu dem sie in die Warteschlange eingereicht wurde. (Dies ist über die Verwendung einer Warteschlangenregel möglich, auf die wir später eingehen werden.) Wird eine Nachricht zwischen WS-Managern ausgetauscht, kann ihr ein Feld hinzugefügt werden, das angibt, dass es sich hier um eine erneute Datenübertragung handelt. Es gibt noch eine ganze Reihe weiterer Felder, die dem Nachrichtenobjekt hinzugefügt werden können.

Bei typischen anwendungsbasierten Nachrichten handelt es sich um Exemplare eines "Erben" der Basisnachrichtenobjektklasse, die je nachdem über zusätzliche Felder verfügen (z. B. für die Rechnungsnummer). Bei einigen dieser Felder handelt es sich um generische Felder, die eine ganze Reihe von Anwendungen gemeinsam haben, wie beispielsweise den Namen des WS-Managers, an den die Antwort gesendet werden soll. Daher unterstützt MQSeries Everyplace Felder, deren Anwesenheit in Nachrichten zu erwarten ist. Dazu gehören unter anderem folgende Felder:

Tabelle 7. Nachrichtenobjektfelder, die unterstützt werden

Feldname	Verwendung
Action (Aktion)	Für die Verwaltung; weist z. B. auf Abfrage-, Erstell- und Löschvorgänge hin.
Correlation ID (Korrelations-ID)	Bytefolge, über die eine Antwort der ursprünglichen Nachricht zugeordnet wird.
Errors (Fehler)	Für die Verwaltung; in diesem Feld werden Fehlerinformationen zurückgegeben.
ExpireTime (Ablaufzeit)	Die Zeitspanne, nach der eine (nicht zugestellte) Nachricht gelöscht wird.
Lock ID (Sperr-ID)	Der Schlüssel, der für die Freigabe der Nachricht erforderlich ist.
Message ID (Nachrichten-ID)	Eine eindeutige Kennung einer Nachricht.
Originating queue manager (Quellen-WS-Manager)	Der Name des WS-Managers, von dem die Nachricht gesendet wurde.
Parameters (Parameter)	Für die Verwaltung; in diesem Feld werden verwaltungsspezifische Informationen übergeben.
Priority (Priorität)	Gibt die relative Priorität bei der Nachrichtenübertragung an.
Reason (Ursachencode)	Für die Verwaltung; in diesem Feld werden Fehlerinformationen zurückgegeben.
Reply-to queue (Warteschlange für Antwortnachrichten)	Der Name der Warteschlange, an die die Antwort gesendet werden soll.

Table 7. Nachrichtenobjektfelder, die unterstützt werden (Forts.)

Feldname	Verwendung
Reply-to queue manager (WS-Manager für Antworten)	Der Name des WS-Managers, an den die Antwort gesendet werden soll.
Resend (Übertragungswiederholung)	Gibt an, dass es sich um die erneute Übertragung einer Nachricht handelt.
Return code (Rückkehrcode)	Für die Verwaltung; in diesem Feld wird der Status eines verwaltungsspezifischen Vorgangs zurückgegeben.
Style (Befehlsart)	Unterscheidet Befehle von Anforderungen/Antworten usw.
Wrap message (Nachricht verpacken)	Gibt an, dass Nachrichten verpackt werden sollen, um die Nachrichtendaten zu schützen.

In allen Fällen steht eine definierte Konstante zur Verfügung, über die der Feldname in einem einzigen Byte übertragen werden kann. Einige Felder werden umfassender unterstützt. Beispielsweise beeinflusst die Priorität (falls angegeben) die Reihenfolge, in der die Nachrichten übertragen werden, die Korrelations-ID bewirkt die Indexierung einer Warteschlange für diese Feldwerte, um einen schnellen Abruf zu ermöglichen, die Ablaufzeit bewirkt das Ablaufen einer Nachricht usw.

In der Verwaltungsanzeige in Abb. 4 auf Seite 24 ist die Struktur einer Nachricht veranschaulicht. Sie sollten beachten, dass durch die Verwendung verschachtelter Felder Nachrichten äußerst komplexe Datenhierarchien enthalten können und dass die Verwaltungs-Tools (und die Anwendungen) die selbstbeschreibende Struktur durchdringen können, sofern dies nicht über Sicherheitseinstellungen verhindert wird.

Nachrichtenobjekte verwenden bei der Serialisierung in oder aus einem Bytebereich für die Übertragung bzw. das Speichern in einer Warteschlange die grundlegenden Ausgabe- und Wiederherstellungsmethoden. Um möglichst wenig Ressourcen zu belegen, wird die zugeordnete Klassendefinition dabei nicht übertragen. Werden die Ausgabe- und Wiederherstellungsmethode durch standardmäßige Serialisierungsmethoden ersetzt, wird die Klassendefinition bei Bedarf übertragen.

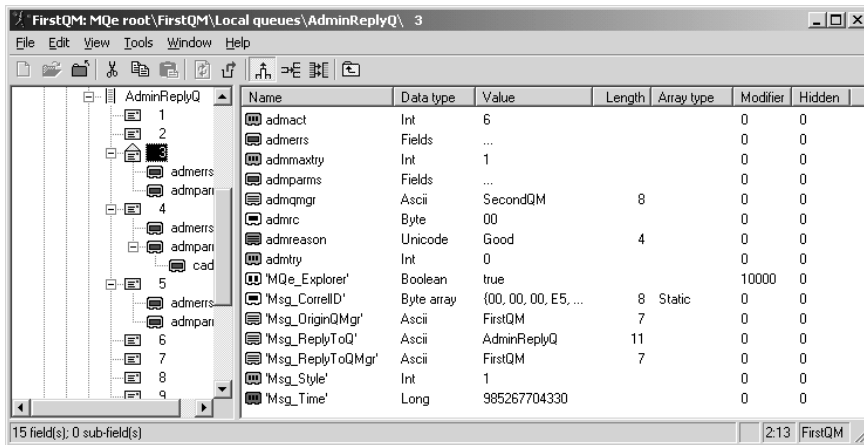


Abbildung 4. Im MQE_Explorer angezeigte Nachrichtenstruktur

In der Regel steht die Klasse jedoch in jedem WS-Manager zur Verfügung, in dem ein Exemplar des Nachrichtenobjekts erstellt werden soll. Muss ein Exemplar eines Exemplars erstellt werden (z. B. zum Zwischenspeichern während der Übertragung), und es ist bekannt, dass die Klassendatei nicht verfügbar ist, kann die Nachricht in eine andere Klasse verpackt werden. Dabei handelt es sich in der Regel um die standardmäßige Nachrichtenobjektklasse. Dieses Verfahren ist auch hilfreich, wenn die Nachricht über ein Attributobjekt geschützt wird. Ist ein solches Attributobjekt vorhanden, ist ein Zugriff auf den Nachrichteninhalt ohne die entsprechenden Sicherheitsschlüssel nicht möglich. Durch Verpacken einer solchen Nachricht bleibt die Nachricht selbst geschützt, sie kann jedoch beliebig ausgegeben und wiederhergestellt werden.

Die Standardmethode des Nachrichtenobjekts für die Ausgabe wurde optimiert, um die Länge der generierten Bytefolge möglichst gering zu halten und so eine effizientere Nachrichtenspeicherung und -übertragung zu erreichen.

Format der Ausgabedaten

Bei der Datenausgabe im Standardformat werden die Felder wie folgt codiert: {Länge Kennung Grenze {Daten}} {Länge Kennung Grenze {Daten}} { ... }

Dabei gilt:

- *Daten*: Der Datenwert. Ganze Zahlen werden komprimiert und alle führenden Nullen und 'F'-Werte entfernt. Booleschen Werten sind keine Datenbytes zugeordnet.
- *Grenze*: Ein besonderes Byte, das die Grenze zwischen der Kennung und dem optionalen Datenelement kennzeichnet. Dieses Byte kennzeichnet darüber hinaus auch den Datenelementtyp.

- *Kennung*: Der Feldname in Form einer ASCII-Bytefolge mit variabler Länge, deren Ende mit einem Endebyte gekennzeichnet ist.
- *Länge*: Die Länge des Datenfeldes. Diese wird durch 1 bis 4 Byte angegeben. Im ersten Byte sind die ersten beiden Bits für Längenangabe des Datenfeldes reserviert. Es werden Längenangaben zwischen 0 und 1.073.741.823 unterstützt.

Daraus ergibt sich ein hochkomprimierter Datenstrom. Eine weitere Komprimierung kann durch die Datenkomprimierung erreicht werden. Die XOR-Komprimierung mit einem vorangehenden Bytestrom sieht vielversprechend aus; da die Felder jedoch variabel sind und sich ihre Reihenfolge ändern kann, kann mit einer einfachen XOR-Verknüpfung nicht immer das gewünschte Ergebnis erreicht werden. In MQSeries Everyplace steht deswegen eine intelligente XOR-Verknüpfung auf Feldebasis zur Verfügung, mit der eine bessere Komprimierung erreicht werden kann.

Warteschlangen

Warteschlangen werden in der Regel zur temporären Speicherung von Nachrichtenobjekten verwendet, die anschließend von Anwendungsprogrammen abgerufen werden. Wie Nachrichten werden auch Warteschlangen aus den Feldobjekten abgeleitet. Ein direkter Zugriff von Anwendungen auf das Warteschlangenobjekt ist in der Regel nicht erlaubt. Stattdessen fungiert der WS-Manager als Mittler zwischen Anwendungsprogrammen und Warteschlangen. Warteschlangen werden über ihren Namen gekennzeichnet, bei dem es sich um eine ASCII-Zeichenfolge mit unbegrenzter Länge handeln kann³, der jedoch innerhalb eines bestimmten WS-Managers eindeutig sein muss. MQSeries Everyplace unterstützt verschiedene Warteschlangenarten:

Lokale Warteschlangen

Lokale Warteschlangen werden von Anwendungen zur sicheren Nachrichtenspeicherung verwendet. Die gespeicherten Nachrichten werden über einen Adapter im permanenten Speicher abgebildet; dabei wird jeweils die gesamte Warteschlange abgebildet. In MQSeries Everyplace stehen eine Reihe von Adaptern zur Verfügung; darüber hinaus können eigene Adapter erstellt oder Adapter Dritter verwendet werden. Bei dem Standardadapter handelt es sich um **MQeDiskFieldsAdapter**, der Warteschlangen im lokalen Dateisystem abbildet und eine gesicherte Zustellung ermöglicht. Ein anderer Adapter (**MQeReduced-DiskFieldsAdapter**) bildet ebenfalls Warteschlangen im Dateisystem ab; dieser Adapter nimmt allerdings Abstriche an der Geschwindigkeit zu Gunsten der Sicherheit in Kauf, damit das Betriebssystem Zeit hat,

3. Um die Interoperabilität sicherzustellen, sollten die MQSeries-Namenseinschränkungen beachtet werden, u. a. auch die maximal zulässige Länge von 48 Zeichen. Die Länge kann auch durch das von Ihnen verwendete Dateisystem eingeschränkt werden.

Warteschlangen

den Inhalt der Pufferspeicher in das physische Plattendateisystem zu übertragen. Ein weiterer Adapter (**MQeMemoryFieldsAdapter**) bildet Warteschlangen im Speicher ab. Dieser Adapter arbeitet am schnellsten, allerdings sind die Nachrichten nach einem Neustart des Betriebssystems oder des WS-Managers nicht mehr vorhanden. Über die Erstellung des entsprechenden Adapters können Nachrichten überall, Warteschlange für Warteschlange, gespeichert werden. So können Nachrichten beispielsweise in einer relationalen Datenbank oder auf einer beschreibbaren CD gespeichert werden. Es gibt Adapter, die das gespiegelte Dateisystem auf dem IBM 4690 Retail Store-Controller oder DB2 für das Speichern von Warteschlangen nutzen.

Lokale Warteschlangen können online oder offline verwendet werden, d. h. mit oder ohne Netzverbindung. Darüber hinaus können für Warteschlangen auch Sicherheitsattribute gesetzt werden, die einen ähnlichen Schutz ermöglichen wie das Attributobjekt für Nachrichten und Feldobjekte. Sicherheitspezifische Überlegungen zu Warteschlangen finden Sie unter „Sicherheit“ auf Seite 52. Der Zugriff auf Nachrichten in lokalen Warteschlangen erfolgt immer synchron, d. h., die Anwendung setzt die Verarbeitung erst fort, nachdem MQSeries Everyplace die Steuerung nach dem Put- bzw. Get-Vorgang zurückgegeben hat.

Ferne Warteschlangen

Bei fernen Warteschlangen handelt es sich um lokale Verweise auf Warteschlangen, die sich in einem fernen WS-Manager 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 Merkmale, wie beispielsweise den Zugriffsmodus (synchron oder asynchron) sowie über sicherheitsspezifische Merkmale und Übertragungsoptionen.

MQSeries Everyplace kann die fernen Warteschlangen automatisch einrichten. Bei einem Versuch, auf eine Warteschlange in einem anderen WS-Manager zuzugreifen (um beispielsweise eine Nachricht zu versenden), überprüft MQSeries Everyplace zunächst, ob eine Definition der fernen Warteschlange vorhanden ist. Ist dies der Fall, wird diese verwendet; ist jedoch keine Definition vorhanden, wird die Warteschlange ermittelt. Dabei werden die Merkmale (Authentifizierung, Verschlüsselung und Komprimierung) ermittelt und eine Definition der fernen Warteschlange erstellt. Diese Form der *Warteschlangenerkennung* ist nur erfolgreich, wenn die Zielwarteschlange verfügbar ist. Ist dies nicht der Fall, muss auf einem anderen Wege eine ferne Definition erstellt werden. Erfolgt eine solche Warteschlangenerkennung,

setzt MQSeries Everyplace den Zugriffsmodus auf 'synchron', da bekannt ist, dass die Warteschlange synchron verfügbar ist.

Der Zugriff auf ferne *synchrone* Warteschlangen ist nur möglich, wenn eine Netzverbindung mit einem Kommunikationspfad zum zugeordneten WS-Manager besteht. Besteht keine Netzverbindung, sind keine Put-, Get- und Browse-Vorgänge möglich; beim Versuch, einen dieser Vorgänge auszuführen (siehe „WS-Managervorgänge“ auf Seite 40), wird eine Ausnahmebedingung ausgegeben. Die Zugriffsberechtigungen und Sicherheitsanforderungen für den Zugriff auf die Warteschlange werden von der tatsächlichen Warteschlange im Ziel-WS-Manager gesteuert. Fehlerbehebung oder Wiederholungen beim Senden bzw. Empfangen von Nachrichten müssen von der Anwendung übernommen werden, da MQSeries Everyplace in diesem Fall nicht mehr für die einmalige gesicherte Zustellung verantwortlich ist.

Bei fernen *asynchronen* Warteschlangen handelt es sich um Warteschlangen, die Nachrichten an ferne Warteschlangen senden, jedoch keine Nachrichten abrufen können. Ist eine Netzverbindung vorhanden, werden die Nachrichten an den zugeordneten WS-Manager und die Warteschlange gesendet. Ist keine Netzverbindung vorhanden, werden die Nachrichten lokal gespeichert und übertragen, sobald eine Netzverbindung hergestellt wurde. Dadurch können Anwendungen auf die Warteschlange zugreifen, auch wenn die Einheit offline ist. Daher verfügen diese Warteschlangen über einen Adapter, der sie in einem Nachrichtenspeicher abbildet, so dass die für eine Übertragung anstehenden Nachrichten temporär im sendenden WS-Manager gespeichert werden können.

Warteschlangen zum Speichern und Weiterleiten

In diesen Warteschlangen werden Nachrichten gespeichert, die an einen oder mehrere Ziel-WS-Manager gerichtet sind. Diese Warteschlangenart hat zwei Hauptfunktionen: Sie ermöglicht zum einen, dass Nachrichten im Netz zwischengespeichert und so etappenweise an ihr Ziel weitergeleitet werden können. Zum anderen können in ihr Nachrichten für den Abruf gespeichert werden (siehe auch **Ausgangsserver-Warteschlangen**).

Warteschlangen zum Speichern und Weiterleiten werden einer Gruppe von WS-Managernamen zugeordnet, für die sie Nachrichten enthalten. Diese WS-Manager werden als Ziel-WS-Manager bezeichnet. Nachrichten, die an einen dieser Ziel-WS-Manager gerichtet sind, werden stattdessen in die entsprechende Warteschlange zum Speichern und Weiterleiten eingereiht. Optional kann für diese Warteschlange auch der Name eines Weiterleitungs-WS-Managers gesetzt werden. In

Warteschlangen

diesem Fall wird die Warteschlange alle Nachrichten an diesen WS-Manager übergeben. Andernfalls speichert die Warteschlange die Nachrichten lediglich.

Diese Warteschlange wird in der Regel (jedoch nicht notwendigerweise) auf einem Server oder Gateway definiert. Ein einzelner WS-Manager kann mehrere Warteschlangen zum Speichern und Weiterleiten enthalten, für jede muss jedoch ein eigener Ziel-WS-Manager angegeben werden. Anwendungsprogramme haben keinen Zugriff auf den Inhalt dieser Warteschlangen. Ebenso ist das Vorhandensein bzw. die Funktion dieser Warteschlangen bei der Nachrichtenübertragung für Anwendungen, die Nachrichten senden, transparent.

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 dagegen ermöglichen den Abruf von Nachrichten aus einer fernen Warteschlange (Nachrichtenempfang im Pull-Modus). Die Definition einer Ausgangsserver-Warteschlange gibt eine Warteschlange zum Speichern und Weiterleiten auf einem fernen WS-Manager an. Die Ausgangsserver-Warteschlange ruft anschließend alle Nachrichten, die an Ausgangsserver-Warteschlangen im lokalen WS-Manager gerichtet sind, 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.

Ausgangsserver-Warteschlangen befinden sich normalerweise in einer Einheit und sind so konfiguriert, dass sie Nachrichten aus einem Server abrufen, sobald für die Einheit eine Netzverbindung hergestellt wird. Nachrichten, die die Ausgangsserver-Warteschlange vom Server im Pull-Modus empfängt, werden in die entsprechende lokale Zielwarteschlange eingereiht. Daher verfügt die Ausgangsserver-Warteschlange über keine Nachrichten, auf die Anwendungen direkt Zugriff haben. Im Hinblick auf den Netzbetrieb ist dieses Abrufverfahren von Nachrichten aus dem Server unter Umständen effizienter als die Nachrichtenübertragung durch den Server im Push-Modus. Für die Ausgangsserver-Warteschlange dient die Bestätigung für die erste Nachricht als Anforderung, die nächste Nachricht (falls vorhanden) abzurufen; würden die Nachrichten hingegen vom Server im Push-Modus gesendet werden, wäre für jede Nachricht eine Antwort/Anforderung für die Übertragung und eine Antwort/Anforderung für die Bestätigung erforderlich. Für Ausgangsserver-Warteschlangen wird in der Regel ein Abfrageintervall gesetzt, so dass der Server in regelmäßigen Abständen auf vorhandene Nachrichten abgefragt wird,

solange eine Netzverbindung besteht. Bei diesem Abfrageintervall handelt es sich um eine Konfigurationsoption für die Verwaltung. Ausgangsserver-Warteschlangen haben eine wichtige Funktion, da sie Clients den Empfang von Nachrichten über Client/Server-Kanäle ermöglichen. Client/Server-Verbindungen beruhen auf dem Prinzip, dass Server keine Datenübertragung einleiten können.⁴

Die Nachrichten in Ausgangsserver-Warteschlangen sind für Anwendungen nicht verfügbar.

Verwaltungswarteschlangen

Über Verwaltungswarteschlangen werden WS-Manager und die ihnen zugeordneten Objekte entweder lokal oder fern konfiguriert. Nachrichten, die an eine Verwaltungswarteschlange gesendet werden, werden von der entsprechenden Verwaltungsnachrichtenklasse verarbeitet; anschließend wird bei Bedarf eine Antwort an die Anwendung zurückgegeben, von der die Nachricht stammt. Weitere Informationen hierzu finden Sie unter „Verwaltung“ auf Seite 44.

MQSeries-Brückenschlangen

Hierbei handelt es sich um eine Sonderform der fernen Warteschlange, bei der sich die Warteschlangendefinition auf einem Gateway und die Zielwarteschlange auf einem MQSeries-WS-Manager befindet. Diese Warteschlangenart stellt eine Schnittstelle zwischen MQSeries Everyplace- und MQSeries-Umgebungen dar. Eventuell erforderliche Daten- oder Nachrichtenumformatierungen werden von Umsetzungsprogrammen vorgenommen. In MQSeries Everyplace steht ein Basisumsetzungsprogramm zur Verfügung; dieses Programm muss von den Programmierern entsprechend den Gegebenheiten angepasst werden.

MQSeries Everyplace speichert Daten sicher in Warteschlangen und sorgt je nach Adapter dafür, dass Nachrichten physisch auf den Datenträger geschrieben und nicht nur vom Betriebssystem temporär gespeichert werden. MQSeries Everyplace protokolliert jedoch keine Änderungen an individuellen Nachrichten und Warteschlangen. Für die Wiederherstellung nach einem Datenträgerausfall sind Hardwarelösungen erforderlich, wie beispielsweise die Verwendung von RAID-Plattensystemen. Ebenso besteht die Möglichkeit, Warteschlangen in einem wiederherstellbaren Speicher, wie beispielsweise in bestimmten Datenbanksystemen, abzubilden.

4. Als Alternative können Peer-zu-Peer-Kanäle verwendet werden, oder beide WS-Manager können jeweils mit Client- und Serverfunktionen konfiguriert werden.

Warteschlangen

In MQSeries Everyplace ist es nicht erforderlich, dass ein WS-Manager über definierte Warteschlangen verfügt. Es werden jedoch bei Bedarf vier Systemwarteschlangen unterstützt:

- **AdminQ:** Diese Warteschlange ist für den Empfang von Verwaltungsnachrichten erforderlich.
- **AdminReplyQ:** Diese Warteschlange kann optional für den Empfang von Antworten auf Verwaltungsnachrichten verwendet werden.
- **DeadLetterQ:** In dieser Warteschlange werden nicht zustellbare Nachrichten gespeichert.
- **SYSTEM.DEFAULT.LOCAL.QUEUE:** Diese Warteschlange hat denselben Namen wie eine obligatorische Systemwarteschlange auf MQSeries-Servern.

Die Warteschlangenmerkmale sind in der folgenden Tabelle aufgeführt. Allerdings gelten nicht alle Merkmale für alle Warteschlangenarten:

Tabelle 8. Warteschlangenmerkmale

Merkmale	Beschreibung
Admin_Class	Warteschlangenklasse
Admin_Name	Warteschlangenname im ASCII-Format
Queue_Active	Gibt an, dass die Warteschlange aktiv ist
Queue_AttRule	Die Regelklasse, die die sicherheitsspezifischen Vorgänge steuert
Queue_Authenticator	Authentifizierungsfunktionsklasse
Queue_BridgeName	Eigenername der MQSeries-Brücke
Queue_ClientConnection	Name der Client-Verbindung
Queue_CloseIdle	Übertragungsfunktion wird nach der Übertragung aller Nachrichten beendet
Queue_CreationDate	Erstellungsdatum der Warteschlange
Queue_Compressor	Komprimierungsfunktionsklasse
Queue_Cryptor	Verschlüsselungsfunktionsklasse
Queue_CurrentSize	Anzahl der Nachrichten in der Warteschlange
Queue_Description	Unicode-Beschreibung
Queue_Expiry	Ablaufzeit für Nachrichten
Queue_FileDesc	Speicherposition und Adapter für die Warteschlange
Queue_MaxMsgSize	Gibt die maximal zulässige Nachrichtenlänge für die Warteschlange an
Queue_MaxQSize	Gibt die maximal zulässige Anzahl von Nachrichten an

Tabelle 8. Warteschlangenmerkmale (Forts.)

Merkmale	Beschreibung
Queue_Mode	Gibt an, ob der Zugriffsmodus synchron oder asynchron ist
Queue_MQMgr	Der Proxy des MQSeries-WS-Managers
Queue_Priority	Die Priorität der Nachrichten (wird durch Angabe eines Nachrichtenwertes außer Kraft gesetzt)
Queue_QAliasNameList	Alternative Namen für die Warteschlange
Queue_QMgrName	Name des WS-Managers, der Eigner der eigentlichen Warteschlange ist
Queue_QMgrNameList	Ziel-WS-Manager
Queue_RemoteQName	Name des fernen MQSeries-Feldes
Queue_Rule	Die Regelklasse für Warteschlangenvorgänge
Queue_QTimerInterval	Verzögerung, mit der anstehende Nachrichten verarbeitet werden
Queue_TargetRegistry	Die Art der Zielregistrierungsdatenbank
Queue_Transporter	Übertragungsfunktionsklasse
Queue_TransporterXOR	Übertragungsfunktion verwendet XOR-Komprimierung
Queue_Transformer	Umsetzungsprogrammklasse

Verwaltungsfunktionen werden zum Erstellen und Löschen von Warteschlangen sowie zum Abfragen bzw. Ändern ihrer Merkmale verwendet.

In den folgenden Bildschirmanzeigen des MQe_Explorer sind zwei der vier Merkmalregisterkarten mit den Merkmalen einer lokalen Warteschlange abgebildet. Inaktivierte Felder deuten darauf hin, dass das betreffende Merkmal für eine Warteschlange dieser Klasse nicht relevant ist.

Warteschlangen

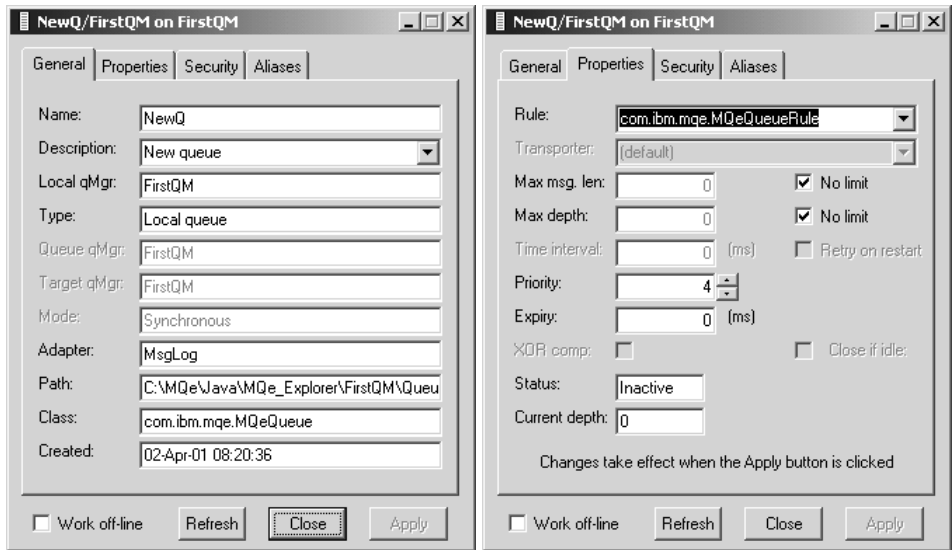


Abbildung 5. Merkmale lokaler Warteschlangen - Anzeige im MQe_Explorer

Warteschlangen können nicht nur zum Speichern von Nachrichten eingesetzt werden. Durch Unterklassen erweiterte Warteschlangen können in Szenarios mit Prozesssteuerungsanwendungen eingesetzt werden, beispielsweise zur direkten Steuerung eines Ventils. So würde bei einer Nachricht der entsprechenden Klasse das Ventil geöffnet, der Durchfluss geändert werden usw. Dabei wäre es nicht die Anwendung, die die Nachrichten abrufen und die Aktion ausführt, sondern die Aktion würde vom Warteschlangenobjekt selbst gesteuert werden. Andere Warteschlangen wiederum könnten für die Aktualisierung von Tabellenkalkulationen oder die Umsetzung von Text in Sprache eingesetzt werden. Dieses Verfahren bietet den Vorteil, dass die Sicherheitseinrichtungen der Warteschlangen und die gesicherte Nachrichtenübertragung weiterhin verfügbar sind. MQSeries Everyplace würde also nach wie vor die einmalige Nachrichtenzustellung sicherstellen, und eine zugeordnete Authentifizierungs- und Verschlüsselungsfunktion würden für die sichere Nachrichtenübertragung durch berechnigte Sender sorgen. Anwendungen erhielten keinen Zugriff auf die Warteschlange, und es wäre keine Zugriffsberechtigung erforderlich.

WS-Manager

Der MQSeries Everyplace-WS-Manager gestattet Anwendungen den Zugriff auf Nachrichten und Warteschlangen und steuert die vorhandenen Kanäle. In MQSeries Everyplace Version 1.2 kann auf einer einzelnen virtuellen Java-Maschine nur jeweils ein WS-Manager aktiv sein. Sind mehrere JVMs vorhanden, können so viele WS-Manager eingesetzt werden wie JVMs vorhanden sind. WS-Manager werden mit Namen gekennzeichnet, bei denen es sich um

systemweit eindeutige Namen und um eine ASCII-Zeichenfolge handeln muss⁵, deren Länge unbegrenzt sein kann.⁶

Ein WS-Manager, der mit der Brückenfunktion konfiguriert und daher in der Lage ist, Nachrichten mit MQSeries-Produkten für Host-Umgebungen und verteilte Umgebungen auszutauschen, wird als Gateway bezeichnet.

WS-Manager können mit oder ohne lokale Warteschlangen konfiguriert werden. Alle WS-Manager unterstützen synchrone Nachrichtenzustellungen; WS-Manager mit lokalen Einreihvorgängen unterstützen auch die asynchrone Nachrichtenzustellung.

Die Wahl einer synchronen oder asynchronen Nachrichtenzustellung hängt von den Warteschlangendefinitionen im sendenden WS-Manager ab. Ist für die Zielwarteschlange die Definition einer fernen Warteschlange mit synchronem Modus vorhanden, so erfolgt die Zustellung synchron. Ist eine Definition einer fernen Warteschlange mit asynchronem Modus vorhanden, erfolgt die Zustellung asynchron, wobei die Definition eine lokale Speicherung der zur Übertragung anstehenden Nachrichten ermöglicht. Ist keine Definition einer fernen Warteschlange, dafür aber eine Warteschlange zum Speichern und Weiterleiten vorhanden, die Nachrichten für den Ziel-WS-Manager handhabt, erfolgt die Zustellung ebenfalls asynchron. In diesem Fall werden die zur Übertragung anstehenden Nachrichten in der Warteschlange zum Speichern und Weiterleiten lokal gespeichert. Ist keine Warteschlangendefinition vorhanden, erfolgt eine *Warteschlangenerkennung*. Ist diese erfolgreich, wird die Nachricht synchron übertragen.

Unabhängig vom synchronen oder asynchronen Modus kann MQSeries Everyplace Nachrichten entweder direkt oder indirekt übertragen; dies hängt von den verfügbaren Verbindungsdefinitionen ab (siehe „Verbindungen“ auf Seite 42). An der direkten Übertragung sind nur zwei WS-Manager beteiligt, der sendende WS-Manager und der Ziel-WS-Manager. Bei der indirekten Übertragung sind eine ganze Reihe von WS-Managern beteiligt, wobei während der Übertragung ein Protokollwechsel stattfinden kann. Bei der indirekten Übertragung im synchronen Modus gibt es keine Unterschiede im Verhalten zur direkten Übertragung. Die zwischengeschalteten WS-Manager stellen lediglich eine Verbindung zwischen Quellen- und Ziel-WS-Manager her. Bei der indirekten Übertragung im asynchronen Modus dagegen können die zwischengeschalteten WS-Manager als Pufferspeicher für die Nachricht auf ihrem Weg vom Quellen- zum Ziel-WS-Manager fungieren. Dieser Einsatz der

5. Dies ist keine Muss-Einschränkung von MQSeries Everyplace oder MQSeries; allerdings können doppelte WS-Managernamen dazu führen, dass Nachrichten an den falschen WS-Manager übertragen werden.

6. Um die Interoperabilität sicherzustellen, sollten die MQSeries-Namenseinschränkungen beachtet werden, u. a. auch die maximal zulässige Länge von 48 Zeichen. Die Länge kann auch durch das von Ihnen verwendete Dateisystem eingeschränkt werden.

WS-Manager hängt davon ab, ob in den zwischengeschalteten WS-Managern Warteschlangen mit ausreichend Speicherplatz in Form von Definitionen ferner Warteschlangen oder Warteschlangen zum Speichern und Weiterleiten definiert wurden.

Zwischen der asynchronen und der synchronen Nachrichtenzustellung bestehen erhebliche Unterschiede: Bei der asynchronen Nachrichtenzustellung übergibt die Anwendung Nachrichten an MQSeries Everyplace, die anschließend an eine ferne Warteschlange übertragen werden. Unmittelbar nach dem Put-Vorgang wird die Steuerung an die Anwendung zurückgegeben. Kann die Nachricht sofort zugestellt oder in einen zwischengeschalteten Puffer eingereiht werden, wird sie gesendet; andernfalls wird sie lokal zwischengespeichert. Die Logik, die für eine erneute Übertragung sorgt, wird von den Regeln gesteuert, die dem WS-Manager und den entsprechenden Warteschlangen zugeordnet sind (siehe „Regeln“ auf Seite 61). Die asynchrone Zustellung sorgt für die einmalige gesicherte Nachrichtenzustellung, da die Nachricht an MQSeries Everyplace übergeben wurde und daher MQSeries Everyplace für die Zustellung zuständig ist.

Bei der synchronen Nachrichtenzustellung reiht die Anwendung die Nachricht in MQSeries Everyplace für deren Übertragung an die ferne Warteschlange ein. MQSeries Everyplace setzt sich synchron mit der Zielwarteschlange in Verbindung und reiht die Nachricht ein. Unmittelbar nach der Zustellung gibt MQSeries Everyplace die Steuerung an die Anwendung zurück. Kann die Nachricht nicht zugestellt werden, wird umgehend eine entsprechende Meldung an die Anwendung geschickt. Im synchronen Modus übernimmt MQSeries Everyplace keine Verantwortung für die Übermittlung der Nachricht.

Definitionen ferner synchroner und asynchroner Warteschlangen können überall im MQSeries Everyplace-Netz eingerichtet werden. Sind im Netz auch MQSeries-WS-Manager und die ihnen zugeordneten Warteschlangen für die Nachrichtenübertragung vorhanden, müssen einige wichtige Einschränkungen beachtet werden.

1. An MQSeries-WS-Manager, die nicht direkt mit einem MQSeries Everyplace-Gateway verbunden sind, ist keine synchrone Nachrichtenübertragung möglich, da MQSeries-Nachrichtenkanäle keine synchrone Übertragung erlauben. Als Ausgleich wird die Definition der synchronen Zustellung geändert, so dass die Zustellung an den MQSeries-WS-Manager erfolgt, der direkt mit dem Gateway verbunden ist. Für die weitere Übertragung wird die asynchrone MQSeries-Nachrichtenübertragung verwendet, damit die Nachricht weiter geschickt werden kann.

2. In MQSeries Everyplace Version 1.2 ist für eine ferne MQSeries-Nachrichtenwarteschlange nur eine ferne synchrone Warteschlangen-Definition möglich. Unter Berücksichtigung der Einschränkung unter (1) bedeutet dies, dass an Warteschlangen in einem MQSeries-WS-Manager, der direkt mit dem Gateway verbunden ist, keine asynchrone Nachrichtenzustellung unter Verwendung dieser Definition möglich ist. Ist die Verwendung von Warteschlangen in diesem Fall erforderlich, besteht jedoch im MQSeries Everyplace-Netz die Möglichkeit, einen zweiten MQSeries Everyplace-WS-Manager einzusetzen, und zwar über entsprechende Definitionen ferner Warteschlangen oder über Warteschlangen zum Speichern und Weiterleiten.

Bei der asynchronen Nachrichtenzustellung übergibt also die lokale Anwendung die Nachricht an MQSeries Everyplace; die Verantwortung für die weitere Übertragung von diesem lokalen WS-Manager aus liegt dann bei MQSeries Everyplace. Daher ist für die sendende bzw. empfangende Anwendung nicht unbedingt eine aktive Netzverbindung erforderlich, d. h. der Zeitpunkt der tatsächlichen Zustellung ist der sendenden Anwendung nicht bekannt. Für die synchrone Nachrichtenzustellung ist eine aktive Netzverbindung erforderlich und die sendende Anwendung weiß, dass die Nachricht an die Warteschlange der Empfängeranwendung übertragen wurde. Sowohl im asynchronen als auch im synchronen Modus muss die Empfängeranwendung nicht verfügbar sein.

MQSeries Everyplace erlaubt keine Auswahl in Bezug auf die einmalige gesicherte Nachrichtenzustellung im asynchronen Modus, wie dies in der MQSeries-Nachrichtenübertragung über die Verwendung permanenter oder nicht permanenter Nachrichten möglich ist. In MQSeries Everyplace kann lediglich auf Warteschlangenebene über die Auswahl des Adapters (zum Speichern der Warteschlangen) festgelegt werden, ob Systemleistung oder Zuverlässigkeit im Vordergrund stehen soll. Beispiel:

- Der Adapter **MQeDiskFieldsAdapter** sorgt dafür, dass die Daten sicher auf Platte geschrieben werden, bevor die Verarbeitung fortgesetzt wird.
- Der Adapter **MQeReducedDiskFieldsAdapter** stellt sicher, dass die Daten an das Betriebssystem übergeben werden (und von dort auch abgerufen werden können), bevor die Verarbeitung fortgesetzt wird.
- Der Adapter **MQeMemoryFieldsAdapter** speichert die Daten im Speicher.

In allen Fällen erfolgt eine einmalige gesicherte Zustellung; ob die Daten wiederhergestellt werden können, hängt jedoch von dem Adapter ab, der ausgewählt wurde. Eine Beschreibung dieser Leistungsabwägungen finden Sie in SupportPac EP01.

WS-Manager

Die WS-Managermerkmale sind in Tabelle 9 aufgeführt.

Tabelle 9. Lokale WS-Manager - Merkmale

Merkmal	Beschreibung
Admin_Class	WS-Managerklasse
Admin_Name	Name des WS-Managers
QMgr_ChnlAttrRules	Kanalattributsregeln
QMgr_ChnlTimeout	Kanalzeitlimit
QMgr_Description	Unicode-Beschreibung
QMgr_QueueStore	Standardmäßige Speicherposition und Adapter für Warteschlangen
QMgr_Rules	Regelklasse für WS-Managervorgänge

In der folgenden Bildschirmanzeige ist eine der zwei Merkmalregisterkarten mit den Merkmalen eines WS-Managers abgebildet:

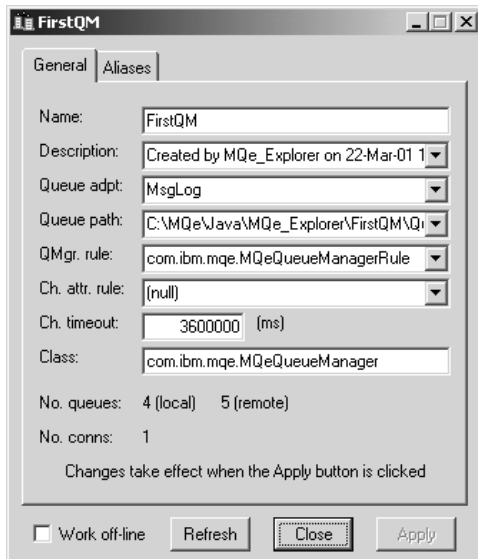


Abbildung 6. WS-Managermerkmale im MQe_Explorer

WS-Managerkonfiguration

Der WS-Manager kommt in einer Umgebung zum Einsatz, die von MQSeries Everyplace vor dem Laden des WS-Managers eingerichtet wird. Die Konfigurationsdaten des WS-Managers werden in dessen Registrierungsdatenbank gespeichert; weitere Informationen hierzu finden Sie unter „Registrierungsdatenbank“ auf Seite 57. Die Warteschlangen selbst (mit den darin enthaltenen Nachrichten) werden in Warteschlangenspeichern gespeichert.

Es gibt verschiedene Möglichkeiten, die MQSeries Everyplace-Umgebung einzurichten: Über Aufrufe über die Anwendungsprogrammierschnittstelle (API, Application Programming Interface), über die im Lieferumfang enthaltenen Dienstprogramme oder über Verwaltungs-Tools wie z. B. den MQE_Explorer. Viele dieser Tools zeichnen die Umgebungsparameter in einer Initialisierungsdatei auf; dies ist jedoch optional. Die wichtigsten Umgebungsparameter sind in der folgenden Tabelle aufgeführt; die Bezeichnungen der Abschnitte, die verwendet werden, entsprechen den Abschnitten in Initialisierungsdateien:

Tabelle 10.

Name des Abschnitts	Merkmal	Beschreibung
[Alias]	(Klassenaliasdefinitionen)	Wenn Klassennamen erforderlich sind, können jederzeit Aliasnamen verwendet werden.
[ChannelManager]	MaxChannels	Gibt die Anzahl der simultanen Client/Server-Kanäle an, die maximal zulässig ist.
[Listener]	Network	Gibt den Adapter und die Nummer des Ports, der auf ankommende Client/Server-Kanalverbindungsanforderungen überwacht werden soll, an.
	Listen	Gibt den Adapter an, der für die Bearbeitung von Verbindungen zuständig ist.
	TimeInterval	Gibt das Zeitlimit für den Client/Server-Kanal an.
[MQBridge]	(Initialisierungsparameter für die Brückenfunktion)	

Tabelle 10. (Forts.)

Name des Abschnitts	Merkmal	Beschreibung
[MQe_Explorer]	(gespeicherte Adressierdaten)	Der MQe_Explorer speichert Informationen darüber, wie ankommende Verbindungen zu dem betreffenden WS-Manager hergestellt werden können.
[Permission]	(zulässige Befehle)	Gibt die zulässigen Kanalbefehle, die Adapterklasse und die Dateideskriptorzuordnungen an.
[PreLoad]	Klassen, die bei der Initialisierung des WS-Managers geladen werden können	Gibt ein Verfahren für das Laden von Anwendungsklassen an.
[QueueManager]	Name	Gibt den Namen des WS-Managers an.
[Registry]	DirName	Gibt das Verzeichnis mit der Registrierungsdatenbank an.
	LocalRegType	Gibt die Art der Registrierungsdatenbank an (Datei oder privat).
	PIN	Gibt die PIN der Registrierungsdatenbank an (oder es wird die Eingabe der PIN gefordert).
	CertReqPIN	Gibt die Zertifikats-PIN an (oder es wird die Eingabe der PIN gefordert).
	KeyRingPassword	Kennwort für Schlüsselring (oder es wird die Eingabe des Kennworts gefordert).
	CAIPAddrPort	IP-Adresse der Zertifizierungsinstanz.

Die folgenden vier Abbildungen zeigen die Registerkarten, die beim Erstellen eines neuen WS-Managers über den MQe_Explorer wichtig sind. Es wird davon ausgegangen, dass keine Umgebung vorhanden ist und keine vorhandene Initialisierungsdatei erforderlich ist. Das Ergebnis ist ein aktiver WS-Manager, dessen Konfigurationsdaten in einer Initialisierungsdatei

gespeichert werden, so dass er einfach durch Öffnen dieser Datei erneut gestartet werden kann.

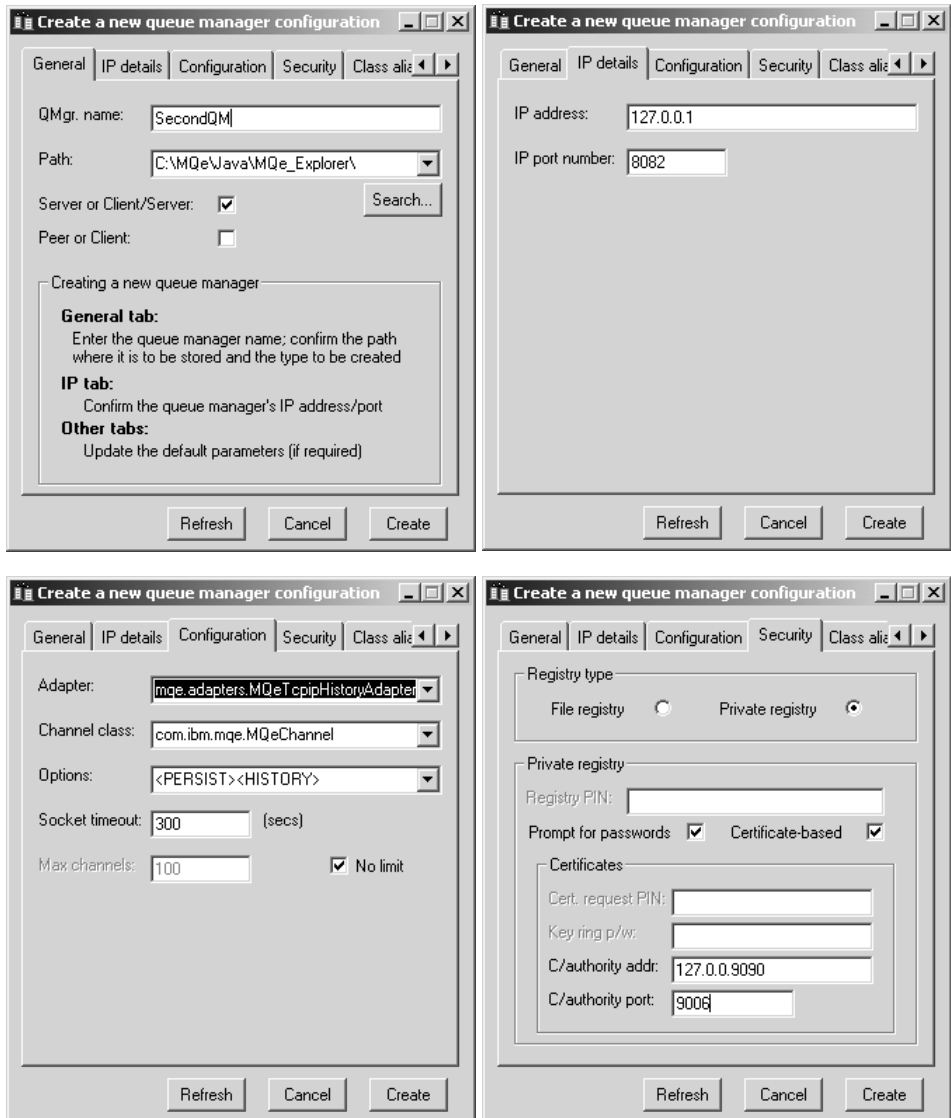


Abbildung 7. Erstellung eines WS-Managers über den MQe_Explorer

Für die Erstellung eines neuen WS-Managers muss die IP-Adresse nicht unbedingt bekannt sein. Der MQe_Explorer erfasst diese Informationen jedoch für den Fall, dass weitere WS-Manager konfiguriert werden, die mit diesem

WS-Manager kommunizieren. Die Port-Nummer ist erforderlich, damit ankommende Client/Server-Verbindungsanforderungen erfasst werden können.

Die Konfigurationsdaten für den Adapter werden vom Kanalempfangsprogramm, die Angaben über das Zeitintervall und die maximal zulässige Anzahl an Kanälen vom Kanalmanager benötigt. Der MQE_Explorer erfasst die anderen Informationen ebenfalls, um die Konfiguration weiterer WS-Manager zu ermöglichen.

In unserem Beispiel wurde eine sichere Registrierungsdatenbank mit einer Authentifizierung, die auf Zertifikaten beruht (siehe „Sicherheit“ auf Seite 52) konfiguriert. PINs und Kennwörter werden inaktiviert, da der Benutzer später bei Bedarf vom MQE_Explorer zu ihrer Eingabe aufgefordert wird.

WS-Managervorgänge

WS-Manager unterstützen Vorgänge im Zusammenhang mit der Nachrichtenübertragung und können optional für die Warteschlangenverwaltung eingesetzt werden. Anwendungen greifen über die Services des WS-Managers auf Nachrichten zu; dabei verwenden sie Methoden wie beispielsweise *get* (Abrufen), *put* (Einreihen), *browse* (Durchsuchen), *wait* (Warten), *listen* (Überwachen) und *delete* (Löschen). Für eine Reihe dieser Vorgänge kann als einer ihrer Parameter ein *Filter* angegeben werden. Bei einem Filter handelt es sich um ein Feldobjekt, das auf Übereinstimmung überprüft wird; alle Felder in der Nachricht können als Filterkriterien für einen selektiven Abruf verwendet werden. Die meisten Methodenaufrufe können darüber hinaus ein Attributobjekt enthalten, das für die Codierung bzw. Decodierung einer Nachricht verwendet wird.

Bei einem Get-Vorgang werden die Nachrichten aus einer Warteschlange entfernt. Je nach festgelegtem Filter werden Nachrichten in Abhängigkeit von ihrer Priorität und außerdem in der Reihenfolge, in der sie in die Warteschlange eingereiht wurden, abgerufen. Weisen also die Nachrichten keine Unterschiede auf, so werden sie nach dem FIFO-Prinzip (First In/First Out) abgerufen. Der Get-Vorgang kann entweder in einem Schritt oder in zwei Schritten ausgeführt werden. Ein Abruf in zwei Schritten erfolgt dann, wenn ein Verlust der Nachricht bei der Übertragung von MQSeries Everyplace an die Anwendung ausgeschlossen werden soll. Dazu wird zunächst ein Get-Aufruf mit einer *Bestätigungs-ID* (die von der Anwendung ausgewählt wird) ausgegeben. Mit diesem Aufruf wird die Nachricht für die Anwendung abgerufen, jedoch nicht sofort aus der Warteschlange gelöscht, sondern verdeckt. Ein anschließender Bestätigungsvorgang unter Angabe der UID der ursprünglichen Nachricht zeigt an, dass der Abruf für die Anwendung erfolgreich war; erst daraufhin wird die Nachricht gelöscht. Schlägt der Abruf fehl, kann die Nachricht wiederhergestellt werden. Put-Vorgänge (Einreihungsvorgänge) verlaufen auf ähnliche Weise.

Durch Angabe der UID können Nachrichten aus einer Warteschlange *gelöscht* werden, ohne abgerufen zu werden.

Sind Lesevorgänge ohne Löschen (Delete) erforderlich, können die Nachrichten in Warteschlangen *angezeigt* (Browse) werden; optional kann dies auch durch Angabe eines Filters erfolgen. Bei diesem Vorgang werden alle Nachrichtenobjekte abgerufen, die mit den Filterkriterien übereinstimmen, sie sind jedoch weiterhin in der Warteschlange vorhanden. *Anzeigen bei gesetzter Sperre* (Browsing under Lock) wird ebenfalls unterstützt. Dabei werden die Nachrichten in der Warteschlange, die mit den Filterkriterien übereinstimmen, gesperrt. Nachrichten können entweder einzeln oder in über einen Filter ausgewählten Gruppen gesperrt werden; vom Sperrvorgang wird eine *Sperr-ID* zurückgegeben. Gesperrte Nachrichten können nur unter Angabe der Sperr-ID abgerufen oder gelöscht werden. Durch Angabe der entsprechenden Option wird bei einem Anzeigevorgang entweder die vollständige Nachricht oder nur die UID zurückgegeben.

Anwendungen können eine angegebene Zeit auf den Empfang einer Nachricht in einer Warteschlange *warten* (Wait). Optional können über einen Filter die gewünschten Nachrichten ausgewählt werden; darüber hinaus ist auch die Angabe einer *Bestätigungs-ID* möglich. Ebenso können Anwendungen auch das Auftreten von MQSeries Everyplace-Nachrichtenergebnissen überprüfen (Listen); auch hier ist die Angabe eines Filters möglich. Empfangsprogramme werden beim Eingang von Nachrichten in einer Warteschlange entsprechend informiert.

In Tabelle 11 ist zu sehen, wie die Warteschlangen für Nachrichtenvorgänge aktiviert werden.

Tabelle 11. Nachrichtenvorgänge in MQSeries Everyplace-Warteschlangen

	Lokale Warteschlange	Ferne Warteschlange ¹	
		Synchron	Asynchron
Browse (±Sperre, ±Filter)	Ja	Ja	
Delete	Ja	Ja	
Get (±Filter)	Ja	Ja	
Listen (±Filter)	Ja		
Put	Ja	Ja	Ja
Wait (±Filter)	Ja	Ja	

Tabelle 11. Nachrichtenvorgänge in MQSeries Everyplace-Warteschlangen (Forts.)

	Lokale Warteschlange	Ferne Warteschlange ¹	
		Synchron	Asynchron
Anmerkungen:			
1. Der synchrone ferne <i>Wait</i> -Vorgang wird über ein Abfrageintervall der fernen Warteschlange implementiert, d. h., die tatsächliche Wartezeit entspricht einem Vielfachen des Abfrageintervalls.			
2. ¹ Die Brücke zwischen MQSeries Everyplace und MQSeries, die in MQSeries Everyplace Version 1.2 zur Verfügung steht, unterstützt nur gesicherte/nicht gesicherte Put-Vorgänge, nicht gesicherte Get-Vorgänge und nicht gesicherte Browse-Vorgänge (ohne Sperre).			

Verbindungen

Die Topologie und der Zugriff über das MQSeries Everyplace-Netz werden über Verbindungsobjekte definiert. Diese Definitionen werden lokal in den einzelnen WS-Managern gespeichert. Sie werden über die standardmäßigen Verwaltungs-Tools erstellt, geändert und gelöscht.

Ein Verbindungsobjekt definiert in der Regel den Zugriff auf einen fernen WS-Manager; daher werden diese Objekte manchmal auch als Definitionen ferner WS-Manager bezeichnet. Die Merkmale sind in der folgenden Tabelle aufgeführt:

Tabelle 12. Verbindungsmerkmale (Merkmale ferner WS-Manager)

Merkmal	Beschreibung
Admin_Name	Name des WS-Managers
Con_Adapter	Der Adapterdateideskriptor
Con_AdapterOptions	Adapteroptionen (z. B. Belegungsprotokoll)
Con_AdapterParm	ASCII-Daten für einen Adapter (z. B. Name eines Servlets)
Con_Aliases	Alternative Namen für den WS-Manager/die Verbindungen
Con_Channel	Die Kanalart für die Verbindung
Con_Description	Unicode-Beschreibung
Queue_QMgrName	Der Eigner der Definition

In den folgenden beiden Bildschirmanzeigen sind zwei der Registerkarten zu sehen, die für die Definition einer Verbindung relevant sind:

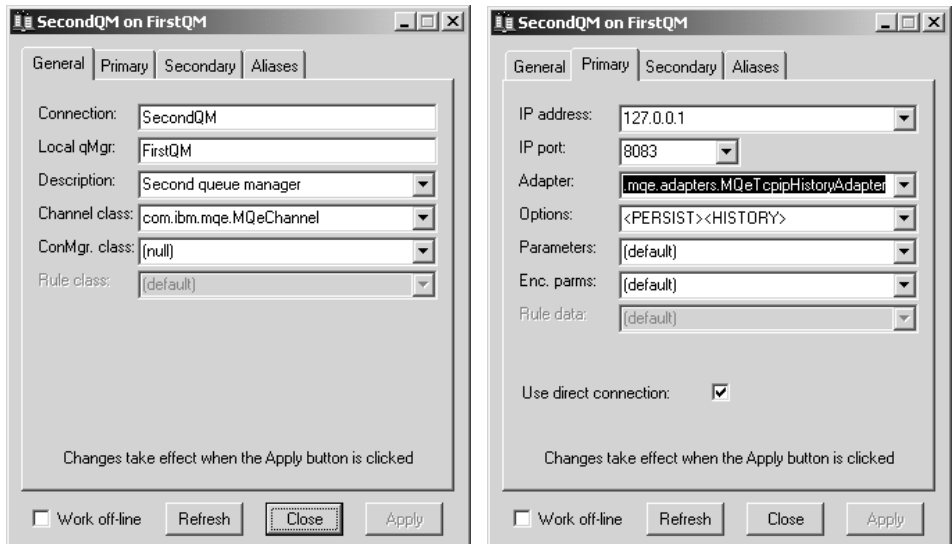


Abbildung 8. Verbindungsdefinition - Anzeige im MQE_Explorer

Daten können in Form von Optionen, Parametern und codierten Parametern an den ausgewählten Kommunikationsadapter übergeben werden.

Es besteht die Möglichkeit, indirekte Verbindungen anzugeben. In diesem Fall leitet MQSeries Everyplace die Verbindung über andere WS-Manager (die miteinander verkettet werden können) weiter; das Übertragungsprotokoll kann sich auf dem Übertragungsweg ändern. Indirekte Verbindungen sind hilfreich, um Einheiten einen einzigen Weg ins MQSeries Everyplace-Netz zu ermöglichen.

Verbindungsobjekte werden außerdem auch für die Definition von Empfangsprogrammen für ankommende Peer-zu-Peer-Kanäle verwendet.

Wie bei den meisten MQSeries-Objekten können für Verbindungen auch Aliasnamen definiert werden. Eine lokale Verbindung (als Verbindung mit einem Namen definiert, der dem des lokalen WS-Managers entspricht) wird für die Definition von Aliasnamen für den lokalen WS-Manager selbst verwendet.

Verwaltung

Die Verwaltung stellt Funktionen zur Konfigurierung und Verwaltung der MQSeries Everyplace-Ressourcen, wie beispielsweise Warteschlangen und Verbindungen, zur Verfügung. Nachrichtenspezifische Funktionen müssen von der Anwendung zur Verfügung gestellt werden. Die Verwaltung wird über eine Schnittstelle aktiviert, die die Erstellung und den Empfang verwaltungsspezifischer Nachrichten handhabt; das zu Grunde liegende Konzept ermöglicht eine identische Handhabung der fernen und lokalen Verwaltung. Anforderungen werden an die Verwaltungswarteschlange auf dem Ziel-WS-Manager gesendet; bei Bedarf können Antworten zurückgegeben werden. Jedes lokale oder ferne MQSeries Everyplace-Anwendungsprogramm kann Verwaltungsnachrichten direkt oder indirekt über Hilfemethoden erstellen und verarbeiten. Verwaltungsnachrichten können außerdem auch indirekt über den MQe_Explorer erstellt werden⁷, ein Verwaltungs-Tool, das eine grafische Schnittstelle für die Systemverwaltung darstellt.

Die Verwaltungswarteschlange selbst weiß nichts über die Ausführung der Verwaltungsvorgänge für die einzelnen Ressourcen; diese Informationen sind in den einzelnen Ressourcen und in der entsprechenden Verwaltungsnachricht enthalten.

Verwaltungsnachrichten

Das MQSeries Everyplace-Nachrichtenobjekt kann durch Verwaltungsnachrichten erweitert werden. In Tabelle 13 sind die Nachrichtenklassen aufgeführt, die für die Verwaltung der MQSeries Everyplace-Ressourcen zur Verfügung stehen. Diese grundlegenden Verwaltungsnachrichten können durch Unterklassen erweitert werden, um so die Verwaltung anderer Objekte zu ermöglichen, beispielsweise kann eine andere Warteschlangenart unter Verwendung einer Unterklasse von **MQeQueueAdminMsg** verwaltet werden. Die MQSeries Everyplace-Brücke zu MQSeries verwendet Unterklassen von **MQeAdminMsg** auf diese Weise.

Tabelle 13. Verwaltungsnachrichtenklassen

Verwaltungsnachrichtenklassen	Verwendung
MQeAdminMsg	Eine abstrakte Klasse, die als Basis aller Verwaltungsnachrichten verwendet wird
MQeQueueManagerAdminMsg	Verwaltung von WS-Managern
MQeQueueAdminMsg	Verwaltung lokaler Warteschlangen
MQeRemoteQueueAdminMsg	Verwaltung ferner Warteschlangen
MQeAdminQueueAdminMsg	Verwaltung der Verwaltungswarteschlange

7. MQe_Explorer gehört nicht zum Lieferumfang von MQSeries Everyplace, kann jedoch kostenlos von der MQSeries Everyplace-Website unter <http://www.ibm.com/software/ts/mqseries/everyplace> heruntergeladen werden.

Tabelle 13. Verwaltungsnachrichtenklassen (Forts.)

Verwaltungsnachrichtenklassen	Verwendung
MQeHomeServerQueueAdminMsg	Verwaltung der Ausgangsserver-Warteschlangen
MQeStoreAndForwardQueueAdminMsg	Verwaltung der Warteschlangen zum Speichern und Weiterleiten
MQeConnectionAdminMsg	Verwaltung der Verbindungen zwischen WS-Managern
MQeClientConnectionAdminMsg	Verwaltung eines Brücken-Client-Verbindungsobjekts, über das die Verbindung zu MQSeries hergestellt wird
MQeListenerAdminMsg	Verwaltung eines Empfangsprozessobjekts für die von der Brücke verwendete Übertragungswarteschlange, mit dem Nachrichten aus MQSeries abgerufen werden
MQeBridgeAdminMsg	Verwaltung einer Brücke zu MQSeries
MQeMQBridgesAdminMsg	Verwaltung einer Liste mit MQSeries-Brücken
MQeMQMgrProxyAdminMsg	Verwaltung der Brückendarstellung eines MQSeries-WS-Managers
MQeMQBridgeQueueAdminMsg	Verwaltung einer MQSeries-Brückenwarteschlange

Die Struktur einer Verwaltungsnachricht hängt von ihrer Klasse ab, d. h. von der Ressource, die sie verwaltet, und dem Vorgang, der für die Ressource ausgeführt wird. Im Allgemeinen haben Verwaltungsnachrichten jedoch die in Tabelle 14 aufgeführte Struktur:

Tabelle 14. Generische Struktur einer Verwaltungsnachricht

Felder der 1. Ebene	Felder der 2. Ebene und darunter	Verwendung
Admin_Action		Erstellen, löschen, abfragen usw.
Admin_Errors		Übergeordnetes Feldobjekt
	Mehrere Felder	Ausführliche Informationen zu jedem Fehler
Admin_MaxAttempts		Gibt an, wie oft der Versuch, einen Verwaltungsvorgang auszuführen, maximal wiederholt werden soll

Tabelle 14. Generische Struktur einer Verwaltungsnachricht (Forts.)

Felder der 1. Ebene	Felder der 2. Ebene und darunter	Verwendung
Admin_Parameters		Übergeordnetes Feldobjekt
	Ressource	Name der zu verwaltenden Ressource
	Mehrere Felder	Ausführliche Parameterdaten für die Nachrichtenklasse und den Vorgang
Admin_Reason		Textnachricht mit der Fehlerursache
Msg_ReplyToQ		Name der Warteschlange, an die die Antwort gesendet werden soll
Msg_ReplyToQMgr		Name des WS-Managers, an den die Antwort gesendet werden soll
Admin_RC		Numerischer Rückkehrcode mit dem Ergebnis
Msg_Style		Befehl oder Anforderung/ Antwort
Admin_TargetQMgr		Name des WS-Managers, der der Eigner der Zielressource ist

Es werden drei Arten von Verwaltungsnachrichten unterstützt, und zwar Befehle (Datagramme), die einen Verwaltungsvorgang angeben, für den keine Antwort erforderlich ist, Anforderungen, für die eine Antwort erforderlich ist, und die Antworten selbst. Die Antwort wird auf der Basis einer Kopie der ursprünglichen Nachricht erstellt; vom Sender können für den Empfänger zusätzliche Felder hinzugefügt werden.

Selektive Verwaltung

Der Verwaltungszugriff kann über die Authentifizierungsfunktion für die Verwaltungswarteschlange gesteuert werden. Lokale Anwendungen werden von der Authentifizierungsfunktion als demselben Benutzer zugehörig eingestuft, d. h., der Verwaltungszugriff wird entweder allen Anwendungen oder keiner erteilt. Ferne Verwaltungsanwendungen werden durch den Aufruf der Authentifizierungsfunktion für den Kanal gesteuert; erst dann ist der Austausch von Verwaltungsnachrichten möglich. Eine Unterscheidung zwischen verschiedenen fernen Benutzern ist möglich, die daher einzeln aktiviert bzw. inaktiviert werden können. In jedem Fall wird der Verwaltungszugriff für jeden Benutzer insgesamt aktiviert oder inaktiviert. Ist eine abgestufte

Verwaltungssteuerung erforderlich, bei der z. B. bestimmten Verwaltungsbenutzern der Zugriff auf einige Warteschlangen erteilt, auf andere verwehrt werden soll, ist eine zusätzliche Programmierung erforderlich. Eine erweiterte Authentifizierungsfunktion verfügt über eine Übersicht über die den verschiedenen Benutzern zugeordneten Berechtigungen; daher können Verwaltungsnachrichten anhand dieser Berechtigungen verarbeitet werden (siehe „Sicherheit“ auf Seite 52). Darüber hinaus können auch die den Warteschlangen zugeordneten Regeln in ähnlicher Weise dazu verwendet werden, bestimmte Vorgänge zuzulassen und andere nicht (siehe „Regeln“ auf Seite 61).

Überwachung und zugehörige Vorgänge

Die Verwaltung geht in der Regel über das bloße Erstellen und Ändern von Objekten hinaus. So gehört zur Verwaltung 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 MQSeries Everyplace über Regeln gesteuert, d. h. Klassen, die bei jeder größeren Änderung am Status eines Objekts oder beim Auftreten bestimmter Fehler aufgerufen werden. In MQSeries Everyplace steht eine Standardgruppe mit Regelklassen zur Verfügung, die jedoch durch benutzerspezifische Klassen ersetzt werden können (siehe „Regeln“ auf Seite 61).

Dynamische Kanäle

Die Kommunikation zwischen MQSeries Everyplace und WS-Managern erfolgt über logische Verbindungen, die so genannten dynamischen Kanäle. Diese unterstützen die bidirektionale Übertragung und werden bei Bedarf vom WS-Manager eingerichtet. Die asynchrone und synchrone Nachrichtenübertragung erfolgt über dieselben Kanäle; hierzu wird ein spezifisches MQSeries Everyplace-Protokoll verwendet. Dieses Protokoll kann auf Nachrichtenbasis angepasst werden, indem das Ausgabeverfahren für die Nachricht geändert wird. In MQSeries hingegen werden in der Regel für den synchronen Austausch Client-Kanäle, und für die bidirektionale asynchrone Nachrichtenübertragung ein Nachrichtenkanalpaar verwendet. Die *Clusternachrichtenkanäle* in MQSeries verfügen über ähnliche Merkmale wie die dynamischen Kanäle in MQSeries Everyplace, es bestehen jedoch eine ganze Reihe wichtiger Unterschiede.

Bei einem dynamischen Kanal handelt es sich um die logische Verbindung zwischen zwei WS-Managern, die für das Senden bzw. Empfangen von Daten hergestellt wird. Zwischen denselben Partnern können mehrere Kanäle gleichzeitig vorhanden sein. Diese verfügen über Merkmale, wie beispielsweise Authentifizierung, Verschlüsselung, Komprimierung und das verwendete Übertragungsprotokoll. Diese Merkmale sind "steckbar" (pluggable)

Dynamische Kanäle

(für unterschiedliche Kanäle können unterschiedliche Versionen verwendet werden); daher verfügt jeder Kanal über eigene Serviceattribute:

- **Authentifizierungsfunktion:** Entweder Null oder ein *Authentifizierungsfunktionsobjekt*, das Benutzer- oder Kanalauthentifizierungen vornehmen kann
- **Kanal:** Die Klasse, die die Übertragungsservices zur Verfügung stellt
- **Komprimierungsfunktion:** Entweder Null oder ein *Komprimierungsfunktionsobjekt*, das Datenkomprimierungen und -dekomprimierungen vornehmen kann
- **Verschlüsselungsfunktion:** Entweder Null oder ein *Verschlüsselungsfunktionsobjekt*, das Datenverschlüsselungen und -entschlüsselungen vornehmen kann
- **Ziel:** Das Ziel des Kanals, z. B. SERVER.XYZ.COM

Die Authentifizierungsfunktion wird in der Regel nur beim Einrichten des Kanals verwendet. Die Komprimierungs- und die Verschlüsselungsfunktionen kommen in der Regel bei jedem Austausch zur Anwendung.

Die einfachste Verschlüsselungsfunktion ist **MQeXorCryptor**, die die zu sendenden Daten über eine XOR-Verknüpfung verschlüsselt. Diese Verschlüsselung ist nicht sicher, jedoch werden die Daten so modifiziert, dass diese nicht angezeigt werden können. Hingegen verwendet **MQe3DESCryptor** die Triple DES-Verschlüsselung. Bei der einfachsten Komprimierungsfunktion handelt es sich um **MQeRleCompressor**, die Daten komprimiert, indem identische Zeichen durch einen Zähler ersetzt werden. In Tabelle 15 auf Seite 53 sind weitere Authentifizierungs-, Komprimierungs- und Verschlüsselungsfunktionen aufgeführt.

Bei der Kanalerstellung werden Protokolladapterspezifikationen verwendet, anhand derer die Verbindungen und Protokolle für einen bestimmten Kanal festgelegt werden. An jedem zwischengeschalteten Knoten werden die Kanaldefinitionen durchsucht, um die für den nächsten Verbindungsabschnitt erforderliche Adresse aufzulösen. Steht keine Definition für die Weiterleitung zur Verfügung, wird der Kanal beendet und alle Nachrichten, die über diesen Kanal gesendet werden, werden an den an diesem Punkt vorhandenen WS-Manager übergeben. Auf diese Weise können zwischengeschaltete Warteschlangen zum Speichern und Weiterleiten sowie Definitionen ferner Warteschlangen genutzt werden.

Kanäle sind für Anwendungsprogrammierer und Administratoren nicht direkt sichtbar und werden vom WS-Manager bei Bedarf erstellt. Kanäle verbinden WS-Manager, und ihre Merkmale werden von MQSeries Everyplace je nach den zu übertragenden Informationen vereinbart bzw. neu vereinbart. Bei Übertragungsprogrammen handelt es sich um MQSeries Everyplace-Komponenten, die unter Verwendung der Kanäle eine Kommunikation

auf Warteschlangenebene ermöglichen. Auch diese Übertragungsprogramme sind für den Anwendungsprogrammierer oder Administrator nicht sichtbar.

Ist eine gesicherte Nachrichtenübertragung erforderlich, werden die Nachrichten nur einmal von MQSeries Everyplace an die Anwendung übertragen. 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 erhalten, kann eine Nachricht noch einmal übertragen werden (einmalige Zustellung ist nicht gleichbedeutend mit einmaliger Übertragung), Nachrichten werden jedoch nicht mehrfach zugestellt.

Adapter

Adapter werden für die Zuordnung von MQSeries Everyplace zu Einheiten-schnittstellen verwendet. Kanäle verwenden die Protokolladapter, um HTTP, natives TCP/IP, UDP und andere Protokolle für die Übertragung einzusetzen. Ähnlich nutzen Warteschlangen die Feldspeicheradapter als Schnittstelle zu Speichersubsystemen wie beispielsweise dem Hauptspeicher oder dem Dateisystem. Adapter stellen für MQSeries Everyplace eine Möglichkeit zur Erweiterung der Einheitenunterstützung und zur Unterstützung der Versionskontrolle zur Verfügung.

Bei dem *Dateideskriptor* handelt es sich um eine Zeichenfolge, über die ein Adapter erkannt, geladen und aktiviert wird.

Verwaltung von Einwählverbindungen

Für die Unterstützung der Netzeinwahl für Einheiten ist das Betriebssystem der Einheit verantwortlich. Versucht MQSeries Everyplace 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 MQSeries Everyplace Version 1.2.

Trace

Ein unabhängiges Programm ist für die Trace-Durchführung zuständig. In MQSeries Everyplace sind Aufrufe für den Trace von Informationsmeldungen, Warnungen und Fehlerbedingungen für das System und den Benutzer integriert. Anwendungen können einen Trace direkt aufrufen und neue Nachrichten hinzufügen bzw. vorhandene Trace-Nachrichten ändern. Das zur Verfügung gestellte Trace-Beispielprogramm ermöglicht die Anzeige ausgewählter Nachrichten, deren Druck oder deren Weiterleitung an das Ereignisprotokoll. Sie können andere Trace-Programme mit zusätzlichen Funktionen oder mit anderen Formatierungs- und Zustellungsmöglichkeiten erstellen.

Der Großteil der MQSeries Everyplace-Ausnahmebedingungen wird zur Verarbeitung an die Anwendung übergeben; die Ausnahmebehandlungsroutine der Anwendung kann diese Ausnahmebedingungen an das Trace-Programm weiterleiten.

Ereignisprotokoll

MQSeries Everyplace stellt Ereignisprotokollmechanismen und -schnittstellen zur Verfügung, mit denen Statusinformationen (z. B. "WS-Manager gestartet") protokolliert werden können. Standardmäßig werden die Protokolleinträge in eine Datei geschrieben; es besteht jedoch die Möglichkeit, die Protokolleinträge abzufangen und an ein anderes Ziel weiterzuleiten. Im MQSeries Everyplace-Ereignisprotokoll werden keine Nachrichtendaten protokolliert, und es kann nicht zur Wiederherstellung von Nachrichten oder Warteschlangen verwendet werden.

Nachrichtenzustellung

Bei MQSeries Everyplace-Netzen handelt es sich um verbundene WS-Manager, die auch Gateways enthalten können. Sie können mehrere physische Netze umfassen, zwischen denen Nachrichten ausgetauscht werden. Im Allgemeinen ermöglichen sie den synchronen und asynchronen Zugriff auf Warteschlangen, und zwar mit einem Programmiermodell, das nicht von der tatsächlichen Position der Warteschlange abhängig ist.

Asynchrone Nachrichtenzustellung

Bei der asynchronen Einreihung einer Nachricht in eine ferne Warteschlange wird das Nachrichtenobjekt lokal im Sicherungsspeicher abgelegt, der für die lokale Definition dieser Warteschlange angelegt wurde; hier werden darüber hinaus auch die Namen des Ziel-WS-Managers und der Zielwarteschlange sowie die Merkmale der Komprimierungs-, Authentifizierungs- und Verschlüsselungsfunktion, die mit denen der Zielwarteschlange übereinstimmen, abgelegt. Die Ausgabemethode (dump) des Objekts wird aufgerufen, wenn das Objekt wie in der Zielwarteschlange definiert in einem sicheren Format im permanenten Speicher abgelegt wird. Die Nachrichtenzustellung wird

vom WS-Manager gesteuert. Er ermittelt für den nächsten Hop einen Kanal mit den entsprechenden Merkmalen (bzw. richtet ihn ein) zum WS-Manager und erstellt anschließend ein Übertragungsprogramm (bzw. verwendet es wieder) für die Übertragung zur Zielwarteschlange. Das Übertragungsprogramm gibt das Objekt in eine Bytefolge aus, die es anschließend überträgt. Der Name des Ziel-WS-Managers und der Warteschlange sind nicht Bestandteil des Nachrichtenflusses.

Falls erforderlich, wird die Nachricht verschlüsselt und komprimiert über den Kanal übertragen. Bei der Ankunft im Ziel-WS-Manager wird sie entschlüsselt und dekomprimiert. Anschließend wird über die Wiederherstellungsmethode der Objektklasse ein neues Nachrichtenobjekt erstellt, das in die Zielwarteschlange eingereiht wird. Wurde die Nachricht nicht an den Ziel-WS-Manager übergeben, wird sie entschlüsselt und dekomprimiert und in eine Warteschlange zum Speichern und Weiterlen mit den entsprechenden Merkmalen für die weitere Übertragung eingereiht. In beiden Fällen wird die Nachricht wie von der Zielwarteschlange vorgegeben in einem sicheren Format in die Warteschlange eingereiht.

Ein Merkmal der asynchronen Nachrichtenzustellung ist die Übergabe der Nachrichten an den WS-Manager auf zwischengeschalteten Hops, wo sie in Warteschlangen für die weitere Übertragung eingereiht werden. Aus diesen zwischengeschalteten Warteschlangen werden Nachrichten erst nach Priorität und an zweiter Stelle erst nach ihrer Zeitmarke weitergeleitet.

Synchrone Nachrichtenzustellung

Die synchrone Nachrichtenzustellung verläuft ähnlich wie die oben beschriebene asynchrone Zustellung, allerdings sind die WS-Manager auf den zwischengeschalteten Hops auf einer wesentlich niedrigeren Ebene (bei der das Übertragungsprogramm und die Kanäle eine Rolle spielen) an dem Vorgang beteiligt. Von Endpunkt zu Endpunkt wird ein Kanal eingerichtet, und zwar unter Verwendung der in der Protokollspezifikation definierten Adapter auf den einzelnen zwischengeschalteten Hops, um den jeweils nächsten Verbindungsabschnitt zu ermitteln. Am Ende des letzten Verbindungsabschnitts, an dem keine weiteren Dateideskriptoren mehr vorhanden sind, wird die Nachricht zur Verarbeitung an die höheren Schichten des WS-Managers übergeben. Der sendende Knoten reiht also die Nachricht nicht selbst in die Warteschlange ein, sondern überträgt sie über einen Kanal über zwischengeschaltete Hops und übergibt sie schließlich dem Ziel-WS-Manager, der die Nachricht in die Zielwarteschlange einreicht.

Die Verbindung zu MQSeries erfolgt über eine Brückenwarteschlange auf dem Gateway, auf dem die Nachricht in ein MQSeries-Format umgesetzt wird. Dieses Verfahren ermöglicht wie in MQSeries Everyplace die synchrone Nachrichtenübertragung von einer Einheit zu MQSeries, wobei der dynamische Kanal am Gateway abgeschlossen wird. Die Nachricht wird in Echtzeit über einen

Nachrichtenzustellung

Client-Kanal vom Gateway an einen MQSeries-Server übertragen. Von dort kann die Zielwarteschlange bei Bedarf die asynchrone Weiterleitung über MQSeries-Nachrichtenkanäle anfordern.

Auf ähnliche Weise kann eine Einheit, auf der lediglich die synchrone Nachrichtenübertragung unterstützt wird, Nachrichten an eine asynchrone MQSeries Everyplace-Warteschlange übertragen, vorausgesetzt, es ist eine entsprechende zwischengeschaltete Stelle vorhanden.

Sicherheit

In MQSeries Everyplace stehen integrierte Sicherheitseinrichtungen zur Verfügung, die den Schutz von Nachrichten im gespeicherten Zustand und bei der Übertragung ermöglichen.

Die MQSeries Everyplace-Sicherheitseinrichtungen können in drei Kategorien eingeteilt werden:

- **Lokale Sicherheit:** Lokaler Schutz von Nachrichtendaten und anderen Daten
- **Sicherheit auf Warteschlangenebene:** Schutz von Nachrichten zwischen dem Ursprungs-WS-Manager und der Zielwarteschlange
- **Sicherheit auf Nachrichtenebene:** Schutz auf Nachrichtenebene für Nachrichten zwischen Sender und Empfänger

Die lokale Sicherheit und die Sicherheit auf Nachrichtenebene werden von MQSeries Everyplace intern genutzt, können jedoch auch MQSeries Everyplace-Anwendungen zugänglich gemacht werden. Bei der Sicherheit auf Warteschlangenebene handelt es sich um einen internen Service.

Die MQSeries Everyplace-Sicherheitseinrichtungen aller drei Kategorien schützen die Nachrichtendaten unter Verwendung eines Attributs (**MQAttribute** oder eine Ableitung davon). Je nach Kategorie wird das Attribut explizit oder implizit verwendet.

Jedes Attribut kann folgende Objekte enthalten:

- Authentifizierungsfunktion
- Verschlüsselungsfunktion
- Komprimierungsfunktion
- Schlüssel
- Name der Zieldefinitionseinheit

Diese Objekte werden je nach MQSeries Everyplace-Sicherheitseinrichtung unterschiedlich verwendet; in allen Fällen wird jedoch der Schutzmechanismus der jeweiligen MQSeries Everyplace-Sicherheitseinrichtung

aktiviert, wenn das einem Nachrichtenobjekt zugeordnete Attribut aufgerufen wird. Dies ist der Fall, wenn die Ausgabemethode einer MQSeries Everyplace-Nachricht aufgerufen wird (wenn die Methode **encodeData** des Attributs verwendet wird, beispielsweise um die Nachrichtendaten zu verschlüsseln und zu komprimieren). Der Schutz der MQSeries Everyplace-Sicherheitseinrichtung wird aufgehoben, wenn die Wiederherstellungsmethode der MQSeries Everyplace-Nachricht verwendet wird (wenn die Methode **decodeData** des Attributs verwendet wird, beispielsweise um die Nachrichtendaten zu dekomprimieren und zu entschlüsseln).

Die in MQSeries Everyplace Version 1.2 für die Authentifizierungs-, Verschlüsselungs- und Komprimierungsfunktion unterstützten Algorithmen sind in Tabelle 15 aufgeführt.

Tabelle 15. Unterstützung für Authentifizierung, Verschlüsselung und Komprimierung

Funktion	Algorithmus
Authentifizierung	WTLS Mini Certificate
	Überprüfung der NT/2000-, AIX- oder Solaris-Identität
Komprimierung	LZW
	RLE
Verschlüsselung	Triple DES
	DES
	MARS
	RC4
	RC6
	XOR

Lokale MQSeries Everyplace-Sicherheit

Lokale Sicherheit schützt die MQSeries Everyplace-Nachrichtendaten (oder **MQeFields** bzw. Erben von **MQeFields**) lokal. Hierzu wird ein Attribut mit einer entsprechenden symmetrischen Verschlüsselungs- und Komprimierungsfunktion erstellt sowie ein entsprechender *Schlüssel* (über ein Kennwort bzw. einen Ausdruck) erstellt und konfiguriert. Dieser Schlüssel wird dem Attribut, und dieses Attribut der MQSeries Everyplace-Nachricht zugeordnet. In MQSeries Everyplace steht für die Konfiguration der lokalen Sicherheitseinrichtungen die Klasse **MQeLocalSecure** zur Verfügung; für die Konfiguration eines entsprechenden Attributs sowie die Verwaltung des Kennworts bzw. Ausdrucks ist jedoch der Benutzer der lokalen Sicherheitseinrichtungen (MQSeries Everyplace intern oder eine MQSeries Everyplace-Anwendung) zuständig.

MQSeries Everyplace-Sicherheit auf Warteschlangenebene

Sicherheit auf Warteschlangenebene ist für synchrone und asynchrone Nachrichten möglich.

Sicherheit auf Warteschlangenebene für synchrone Nachrichten

Bei Verwendung der Sicherheit auf Warteschlangenebene für synchrone Nachrichten können Anwendungen Sicherheitsüberlegungen MQSeries Everyplace überlassen. Warteschlangen verfügen über die Merkmale "Authentifizierung", "Verschlüsselung" und "Komprimierung", mit denen die für den Schutz des Nachrichtenflusses (und der Nachrichten im permanenten Speicher) erforderliche Sicherheitsstufe ermittelt werden kann.

Beim Senden einer Nachricht werden die Sicherheitsmerkmale der Zielwarteschlange aus der lokalen Registrierungsdatenbank abgerufen. Sind keine Sicherheitsmerkmale vorhanden, versucht der WS-Manager, die Merkmale der Zielwarteschlange über den Ziel-WS-Manager zu ermitteln und für die spätere Verwendung im Cache zu speichern. Besteht ein Kanal zu diesem WS-Manager, wird dieser verwendet; andernfalls wird ein neuer Kanal erstellt. Anschließend werden die Attribute der Zielwarteschlange abgerufen.

Je nach der geforderten Servicequalität können die Attribute des Kanals zum Ziel-WS-Manager dynamisch geändert werden. Die möglichen Änderungen hängen von den definierten Regeln ab. Im Allgemeinen lässt eine Regel einen Upgrade der Sicherheitsstufe zu (z. B. von keinem zu einem minimalen Schutz oder von einem minimalen zu einem umfangreichen Schutz). Ist kein Upgrade des Kanals möglich oder wird die Sicherheitsstufe als zu hoch eingeschätzt (wenn z. B. kein Schutz erforderlich ist, der verfügbare Kanal jedoch umfassenden Schutz bietet), wird ein neuer Kanal erstellt. Es steht ein Kanal-Pool (der nach Möglichkeit wieder verwendet wird) mit Merkmalen zur Verfügung, die je nach den Anforderungen des Datenverkehrs dynamisch geändert werden. Kanäle, die nicht mehr erforderlich sind, werden automatisch gelöscht. Nachrichten werden immer in Warteschlangen mit der Sicherheitsstufe eingereiht, die über die Merkmale der Zielwarteschlange definiert ist.

Authentifizierungen werden auf Kanalebene ausgeführt, wobei der Aufwand pro Nachricht so gering wie möglich gehalten wird. Sicherheit auf Warteschlangenebene für die synchrone Nachrichtenübertragung wird in der Regel auch zusammen mit symmetrischen Verschlüsselungsfunktionen eingesetzt, was eine rasche Ver- und Entschlüsselung ermöglicht. In diesen Fällen verwendet MQSeries Everyplace jedoch zu Anfang die asymmetrische RSA-Verschlüsselung, um den Austausch zu schützen, mit dem ein gemeinsam verwendeter Schlüssel auf der Sender- und Empfängerseite eingerichtet wird. Anschließend wird eine symmetrische Verschlüsselung zum Schutz der übertragenen Daten verwendet. Unbefugte Datenzugriffe werden in MQSeries Everyplace außerdem noch durch die dynamische Änderung des Schlüssels für jeden Kanalaustausch erschwert. Darüber hinaus sichert MQSeries Everyplace

die Datenintegrität, indem vor dem eigentlichen Übertragungsvorgang ein Auszug der Daten erstellt und angehängt wird, der auf dem Empfänger wiederhergestellt und überprüft wird.

Sicherheit auf Warteschlangenebene für asynchrone Nachrichten

Anders als bei der oben beschriebenen synchronen Nachrichtenübertragung ist bei der asynchronen Nachrichtenübertragung die Zielwarteschlange zum Zeitpunkt des **putMessage** (Einreihung der Nachricht in die Warteschlange) nicht unbedingt verfügbar. In diesem Fall kann der WS-Manager die Nachricht nicht sofort übertragen, sondern stellt sie in die Übertragungswarteschlange, wobei sie anhand der Merkmale der Zielwarteschlangen verschlüsselt wird. Zum Zeitpunkt der Übertragung wird die Nachricht entschlüsselt und mit den entsprechenden Merkmalen über den Kanal übertragen. Dadurch sind Nachrichten auch dann geschützt, wenn sie für die Übertragung anstehen. Für die asynchrone Nachrichtenübertragung ist die Definition einer fernen Warteschlange erforderlich, damit die Merkmale der Zielwarteschlange ermittelt werden können.

Bei der asynchronen Nachrichtenübertragung ist keine Authentifizierung zwischen Ursprung und Ziel möglich. Ist eine Authentifizierung erforderlich, z. B. damit ein Empfänger den Sender ermitteln kann (um über die Annahme der Nachricht zu entscheiden oder den Urheber eindeutig über die digitale Unterschrift (Non-Repudiation) zu identifizieren) oder ein Sender sicherstellen kann, dass die Nachricht nur vom betreffenden Empfänger verarbeitet werden kann, ist Sicherheit auf Nachrichtenebene erforderlich.

Die Sicherheit auf Warteschlangenebene kann zusammen mit der Sicherheit auf Nachrichtenebene eingesetzt werden, ist jedoch nicht erforderlich, da der Schutz der Nachrichtendaten gewährleistet ist.

Sicherheit auf Nachrichtenebene

Die Sicherheit auf Nachrichtenebene schützt die Nachrichtendaten, die zwischen einer sendenden und einer empfangenden MQSeries Everyplace-Anwendung übertragen werden.

Die Sicherheit auf Nachrichtenebene ist ein Service der Anwendungsschicht; die sendende MQSeries Everyplace-Anwendung muss ein Attribut auf Nachrichtenebene einrichten, das beim Einreihen der Nachricht in eine Zielwarteschlange mit **putMessage** übergeben wird. Die empfangende Anwendung muss ein entsprechendes Attribut auf Nachrichtenebene einrichten und dem empfangenden WS-Manager übergeben; dieses Attribut muss verfügbar sein, wenn die Anwendung die Nachricht unter Aufruf von **getMessage** aus der Zielwarteschlange abrufen.

Wie die lokalen Sicherheitseinrichtungen macht sich auch die Sicherheit auf Nachrichtenebene die Anwendung eines Attributs auf ein Nachrichtenobjekt

zunutze. Der WS-Manager der sendenden Anwendung verwendet beim **putMessage**-Aufruf die Ausgabemethode, wobei die Nachrichtendaten mit der Methode **encodeData** des Attributs geschützt werden. Der WS-Manager der empfangenden Anwendung verwendet beim Aufruf **getMessage** der Anwendung die Wiederherstellungsmethode, bei der die ursprünglichen Nachrichtendaten mit der Methode **decodeData** des Attributs wiederhergestellt werden.

MQSeries Everyplace stellt zwei alternative Attribute für die Sicherheit auf Nachrichtenebene zur Verfügung:

MQeMAttribute

Dieses Attribut wird für die Kommunikation zwischen Unternehmen verwendet, bei der das gegenseitige Vertrauen seine Umsetzung in der Anwendungsschicht findet und kein vertrauenswürdiger Dritter erforderlich ist. Alle verfügbaren symmetrischen MQSeries Everyplace-Verschlüsselungs- und Komprimierungsfunktionen können verwendet werden. Wie bei der lokalen Sicherheit muss der Attributsschlüssel zunächst eingerichtet werden, bevor er mit **putMessage** oder **getMessage** zur Verfügung gestellt wird. **MQeAttribute** stellt einen einfachen, aber wirkungsvollen Mechanismus für den Schutz auf Nachrichtenebene zur Verfügung, wobei die Nachrichtendaten durch eine effiziente Verschlüsselung geschützt werden, ohne dass hierzu eine Infrastruktur, die mit öffentlichen Schlüsseln arbeitet (PKI = Public Key Infrastructure), erforderlich ist.

MQeMTrustAttribute

Bei diesem Attribut handelt es sich um eine erweiterte Lösung, bei der digitale Unterschriften verwendet werden und die sich die standardmäßige öffentliche Schlüsselinfrastruktur zunutze macht. Es setzt die digitale Unterschrift/Überprüfung des Standards ISO9796 ein und ermöglicht damit der empfangenden Anwendung, die Identität des Nachrichtensenders zu überprüfen. Die von diesem Attribut zur Verfügung gestellte Verschlüsselungsfunktion wird zum Schutz der Nachrichtendaten eingesetzt. Ein SHA1-Auszug stellt die Nachrichtenintegrität sicher und die RSA-Verschlüsselung/Entschlüsselung gewährleistet, dass die Nachricht nur vom betreffenden Empfänger wiederhergestellt werden kann. Wie bei **MQeAttribute** können alle verfügbaren symmetrischen MQSeries Everyplace-Verschlüsselungs- und Komprimierungsfunktionen verwendet werden. Um den Umfang so gering wie möglich zu halten, werden WTLS Mini Certificates verwendet. Die Verfügbarkeit der für die Authentifizierung (Unterschriftenüberprüfung) sowie für die Verschlüsselung/Entschlüsselung erforderlichen Informationen an beiden Endpunkten wird über die MQSeries Everyplace-Standardinfrastruktur zur Verfügung gestellt.

Eine über **MQeTrustAttribute** geschützte Nachricht hat in der Regel folgendes Format:

RSA-enc{SymKey}, SymKey-enc {Data, DataDigest, DigSignature}

Dabei gilt:

RSA-enc: RSA-Verschlüsselung mit dem öffentlichen Schlüssel des betreffenden Empfängers

SymKey: Der symmetrische quasi-zufällig generierte Schlüssel

SymKey-enc: Symmetrische Verschlüsselung mit SymKey

Data: Nachrichtendaten

DataDigest: Auszug der Nachrichtendaten

DigSignature Digitale Unterschrift des Senders für die Nachrichtendaten

Die Sicherheit auf Nachrichtenebene und die Sicherheit auf Warteschlangenebene arbeiten unabhängig voneinander.

Registrierungsdatenbank

Bei der Registrierungsdatenbank handelt es sich um den Primärspeicher für WS-Managerspezifische Informationen; für jeden WS-Manager ist jeweils eine Datenbank vorhanden. Jeder WS-Manager speichert folgende Daten in dieser Datenbank:

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

Der Zugriff auf die Registrierungsdatenbank ist in der Regel auf den berechtigten WS-Managerbenutzer beschränkt und durch eine PIN-Nummer geschützt; diese Sicherheitseinschränkungen können jedoch über eine Option für Benutzer aufgehoben werden, für die die Systembelegung einen höheren Stellenwert als die Sicherheit einnimmt.

Authentifizierbare MQSeries Everyplace-Definitionseinheiten

Die Sicherheit auf Warteschlangenebene, bei der eine gegenseitige Authentifizierung über Mini Certificates stattfindet, und die Sicherheit auf Nachrichtenebene, bei der digitale Unterschriften verwendet werden, haben zum Konzept der *authentifizierbaren Definitionseinheit* geführt. Bei der gegenseitigen Authentifizierung denkt man in der Regel an die gegenseitige Überprüfung zweier Benutzer (Personen), aber im Allgemeinen gibt es dieses Konzept von Benutzern bei der Nachrichtenübertragung nicht. Die Nachrichtenübertragung wird in der Regel auf der Anwendungsebene gesteuert, d. h. von den Benutzern

der Nachrichtenübertragungsservices. Daher wurde in MQSeries Everyplace im Hinblick auf das *Ziel einer Authentifizierung* das Konzept vom *Benutzer* durch die abstrakte *authentifizierbare Definitionseinheit* ersetzt. Dies schließt nicht aus, dass es sich bei den authentifizierbaren Definitionseinheiten auch um Personen handeln kann, dies wird jedoch von der Anwendung festgelegt. Intern werden von MQSeries Everyplace alle WS-Manager als authentifizierbare Definitionseinheiten definiert, die Mini Certificate-spezifische Services aufrufen oder für die diese Services aufgerufen werden können. Darüber hinaus werden Warteschlangen, die die Mini Certificate-basierte Authentifizierungsfunktionen verwenden, von MQSeries Everyplace als authentifizierbare Definitionseinheiten definiert. So kann ein WS-Manager, der diese Services unterstützt, über eine authentifizierbare Definitionseinheit, nämlich den WS-Manager, oder über mehrere authentifizierbare Definitionseinheiten, nämlich den WS-Manager und alle Warteschlangen, die die Mini Certificate-basierte Authentifizierungsfunktion verwenden, verfügen.

Private Registrierungsdatenbank und Berechtigungen

Sinnvollerweise muss jede authentifizierbare Definitionseinheit über eigene Berechtigungen verfügen. Daraus ergeben sich zwei Überlegungen, zum einen, wie die Registrierung durchgeführt werden muss, damit die Berechtigungen abgerufen werden können, und zum anderen, wo diese Berechtigungen sicher verwaltet werden können. In der Regel bereitet die Lösung dieser Fragen größere Schwierigkeiten als die Implementierung des Verschlüsselungsverfahrens selbst. MQSeries Everyplace stellt Standard-services zur Verfügung, über die authentifizierbare Definitionseinheit eine automatische Registrierung vornehmen können. Die private Registrierungsdatenbank (abgeleitet von der Basisregistrierungsdatenbank) ermöglicht die gesicherte Verwaltung der privaten Berechtigungen einer authentifizierbaren Definitionseinheit, die öffentliche Registrierungsdatenbank (ebenfalls von der Basisregistrierungsdatenbank abgeleitet) ermöglicht die Verwaltung öffentlicher Berechtigungen. Die private Registrierungsdatenbank bietet die Möglichkeiten der Basisregistrierungsdatenbank mit vielen Merkmalen eines gesicherten oder verschlüsselten Token; so kann sie z. B. als gesichertes Repository für öffentliche Objekte wie Mini Certificates oder für private Objekte wie private Schlüssel verwendet werden. Ein besonderer Mechanismus erlaubt nur dem berechtigten Benutzer den Zugriff auf die privaten Objekte. Die private Registrierungsdatenbank stellt außerdem Unterstützung für Services (z. B. digitale Unterschriften und RSA-Verschlüsselung) 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, kann jedoch später unter Umständen auf die Zuordnung zu übertragbaren Token erweitert werden.

Automatische Registrierung

MQSeries Everyplace stellt Standardservices für die Unterstützung der automatischen Registrierung zur Verfügung. Diese Services werden bei der Konfiguration einer authentifizierbaren Definitionseinheit 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 Registrierungsdatenbank der authentifizierbaren Definitionseinheit abgelegt. Die automatische Registrierung umfasst die Generierung eines neuen RSA-Schlüsselpaares, den Schutz und das Speichern des privaten Schlüssels in der privaten Registrierungsdatenbank sowie die Packung des öffentlichen Schlüssels in die Anforderung für ein *neues Zertifikat* an den standardmäßigen Mini Certificate Server. Ist der Mini Certificate Server konfiguriert und verfügbar, gibt er das neue Mini Certificate für die authentifizierbare Definitionseinheit zusammen mit dem eigenen Mini Certificate zurück. Diese werden zusammen mit dem geschützten privaten Schlüssel als neue Berechtigungen der authentifizierbaren Definitionseinheit in deren privaten Registrierungsdatenbank gespeichert. Während mit der automatischen Registrierung auf einfache Weise die Berechtigungen für eine authentifizierbare Definitionseinheit eingerichtet werden können, ist für den Schutz auf Nachrichtenebene (**MQeTrustAttribute**, siehe oben) außerdem noch der Zugriff auf den öffentlichen Schlüssel (Mini Certificate) des betreffenden Empfängers erforderlich.

Öffentliche Registrierungsdatenbank und Zertifikatreplikation

MQSeries Everyplace stellt Standardservices zur Verfügung, die MQSeries Everyplace-Komponenten die gemeinsame Benutzung der öffentlichen Berechtigungen (Mini Certificates) authentifizierbarer Definitionseinheiten ermöglichen. Diese öffentlichen Berechtigungen sind die Voraussetzung für die Sicherheit auf Nachrichtenebene auf der Basis von **MQeMTrust**. Die öffentliche MQSeries Everyplace-Registrierungsdatenbank stellt ein Repository mit öffentlichen Zugriffsmöglichkeiten für Mini Certificates dar. Dies entspricht einem persönlichen Telefonverzeichnis auf einem mobilen Telefon, nur dass es sich hier nicht um Telefonnummern, sondern um die Mini Certificates der authentifizierbaren Definitionseinheiten handelt, zu denen am häufigsten eine Verbindung hergestellt wird. Bei den Services der öffentlichen Registrierungsdatenbank handelt es sich nicht nur um passive Services. Wird die Datenbank nach einem nicht vorhandenen Mini Certificate durchsucht und ist sie mit einer gültigen Ausgangsserverkomponente konfiguriert, ruft die öffentliche Registrierungsdatenbank das angeforderte Mini Certificate automatisch aus der öffentlichen Registrierungsdatenbank des Ausgangsservers ab. Diese Services können als intelligenter Service für die automatische Replikation von Mini Certificate eingesetzt werden, die dafür sorgen, dass die richtigen Mini Certificates immer zur Verfügung stehen.

Verwendung der Datenbankservices durch Anwendungen

Der MQSeries Everyplace-WS-Manager ist für die Nutzung der privaten und öffentlichen Registrierungsdatenbanken konzipiert, allerdings ist der Zugriff auf diese Services nicht eingeschränkt. Es sind MQSeries Everyplace-Lösungen denkbar, in denen eigene authentifizierbare Definitionseinheiten, beispielsweise Benutzer, definiert und verwaltet werden können. Die Services der privaten Registrierungsdatenbank können in diesem Fall für die automatische Registrierung und die Verwaltung der Berechtigungen verwendet werden, während die Services der öffentlichen Registrierungsdatenbank den Zugriff auf Berechtigungen ermöglichen, wann immer dies erforderlich ist. Alle registrierten authentifizierbaren Definitionseinheiten können als Aufrufer bzw. Empfänger von Services auf Nachrichtenebene, die mit **MQEmTrustAttribute** geschützt werden, verwendet werden.

Standardservices für die Vergabe von Mini Certificates

MQSeries Everyplace stellt Standardservices für die Vergabe von Mini Certificates zur Verfügung, die für die Verarbeitung von Anforderungen für eine automatische Registrierung in privaten Registrierungsdatenbanken konfiguriert werden können. Mit den von MQSeries Everyplace zur Verfügung gestellten Tools kann eine Lösung implementiert werden, mit der ein Service zur Vergabe von Mini Certificates an bestimmte Definitionseinheiten eingerichtet und verwaltet werden kann. Dieser Service verfügt über folgende Merkmale:

- Verwaltung einer Gruppe registrierter authentifizierbarer Definitionseinheiten
- Vergabe von Mini Certificates
- Verwaltung des WAP WTLS Mini Certificate-Repositorys

Mit den in MQSeries Everyplace verfügbaren Tools kann der Administrator der Services zur Vergabe von Mini Certificates die Vergabe von Mini Certificates an bestimmte Definitionseinheiten steuern, indem der Name der jeweiligen Definitionseinheit und deren registrierte Adresse registriert und eine Anforderungs-PIN für die einmalige Verwendung eines Zertifikats definiert werden. Dies erfolgt in der Regel, nachdem die Identität des Anforderers offline überprüft wurde. Die Anforderungs-PIN für das Zertifikat wird an den entsprechenden Benutzer gesendet (z. B. auf ähnliche Weise, wie Bankkarten-PINs bei Ausgabe einer neuen Bankkarte an den Benutzer gesendet werden). Die Benutzer der privaten Registrierungsdatenbank (z. B. die MQSeries Everyplace-Anwendung oder der MQSeries Everyplace-WS-Manager) können anschließend so konfiguriert werden, dass sie beim Start diese Anforderungs-PIN für das Zertifikat zur Verfügung stellen. Wenn von der privaten Registrierungsdatenbank die automatische Registrierung ausgelöst wird, überprüft der Service zur Vergabe von Mini Certificates die betreffende neue Zertifikatsanforderung (wobei der Name der angegebenen Definitionseinheit und die Anforderungs-PIN für das Zertifikat mit den registrierten Angaben

verglichen werden), gibt das neue Mini Certificate aus und setzt die registrierte Anforderungs-PIN für das Zertifikat zurück, damit diese nicht mehr verwendet werden kann. Alle automatisch registrierten Mini Certificate-Anforderungen werden über einen gesicherten Kanal übertragen.

Die von einem entsprechenden Service ausgegebenen Mini Certificates werden in der eigenen Registrierungsdatenbank dieses Services gespeichert. Bei der erneuten Ausgabe eines Mini Certificate (z. B. im Falle einer Zeitlimit-überschreitung) wird das abgelaufene Mini Certificate archiviert.

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 bzw. Entfernen von Klassenaliasnamen
- Adapterdefinitionen
- Zuordnung von Dateideskriptoren
- Verarbeitung von Kanalbefehlen

Anpassung

Regeln

Bei Regeln handelt es sich um Java-Klassen, über die das Betriebsverhalten von MQSeries Everyplace bei verschiedenen Statusänderungen gesteuert werden kann. Die notwendigen Standardregeln werden zur Verfügung gestellt, sie können jedoch vom Kunden bei Bedarf durch anwendungs- oder installationspezifische Regeln ersetzt werden. Die unterstützten Regelarten unterscheiden sich nicht durch ihre Funktion, sondern durch die Auslöseart. Regeln enthalten Logikcode und können daher eine Vielzahl von Funktionen ausführen.

Attributsregeln

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.

MQSeries-Brückenregeln

Diesen Regelklassen wird die Steuerung übergeben, wenn im Code der Brücke zwischen MQSeries Everyplace und MQSeries-Brücke eine Änderung auftritt. Für jede der folgenden Situationen steht eine eigene Brückenregelklasse zur Verfügung:

- Vorgehensweise, wenn eine Nachricht von MQSeries nicht vom Empfangsprogramm an MQSeries Everyplace übergeben werden kann (z. B. weil die Nachricht zu lang oder die Warteschlange nicht vorhanden ist).
- Der Status, in dem von der Brücke verwaltete Objekte unmittelbar nach dem Erstellen eines Serverexemplars gestartet werden sollen.
- Vorgehensweise, wenn die Brücke ein Problem in Zusammenhang mit der Sync-Warteschlange (dem permanenten Speicher, der für die Wiederherstellung nach einem Absturz herangezogen wird) auf MQSeries feststellt. Dieses Problem wird durch die Standardregel lediglich angezeigt.
- Die Vorgehensweise bei der Umsetzung von MQSeries Everyplace-Nachrichten in MQSeries-Nachrichten und umgekehrt. Die Umsetzungsprogramme für die Nachrichtenkonvertierung zwischen MQSeries Everyplace und MQSeries-Nachrichten werden nicht von MQeRule-Klassen abgeleitet, sondern müssen die Schnittstelle MQeTransformerInterface implementieren. Davon abgesehen arbeiten Umsetzungsprogramme ähnlich wie Regeln und werden aufgerufen, wenn für eine Nachricht eine Formatumsetzung erforderlich ist.

Warteschlangenregeln

An diese Regelklasse wird die Steuerung übergeben, wenn eine Änderung am Warteschlangenstatus auftritt, wie beispielsweise in folgenden Fällen:

- Beim Einreihen einer Nachricht in die Warteschlange. In diesem Fall kann beispielsweise festgestellt werden, ob ein Schwellenwert (Anzahl der Nachrichten, Nachrichtenlänge, ungültiges Merkmal) überschritten wurde.
- Es wurden Warteschlangenkenndaten zugeordnet oder geändert.
- Die Warteschlange wird geöffnet bzw. geschlossen.
- Die Warteschlange soll gelöscht werden.

WS-Managerregeln

An diese Regelklasse wird die Steuerung übergeben, wenn eine Änderung am WS-Managerstatus auftritt, wie beispielsweise in folgenden Fällen:

- Der WS-Manager wird geöffnet. Es kann beispielsweise ein Zeitgeber-Thread, der im Hintergrund abläuft, gestartet werden, damit zeitlich ange-setzte Vorgänge aktiviert werden.
- Der WS-Manager wird geöffnet. Beispielsweise kann der Zeitgeber-Thread, der im Hintergrund abläuft, beendet werden.
- Es wird eine neue Warteschlange hinzugefügt.

Verbindungsarten

MQSeries Everyplace kann Client/Server- und Peer-zu-Peer-Betrieb unterstützen. *Clients* können die Kommunikation mit einem Server eröffnen. Ein *Server* hingegen kann nur auf die von einem Client gesendeten Anforderungen reagieren. In einer *Peer-zu-Peer*-Umgebung ist der Kommunikationsfluss in beide Richtungen möglich. Für diese Verbindungsarten sind jeweils unterschiedliche MQSeries Everyplace-Komponenten erforderlich. Hierbei handelt es sich um folgende Komponenten:

- **Kanalempfangsprogramm:** Ist für eingehende Verbindungsanforderungen empfangsbereit.
- **Kanalmanager:** Unterstützt mehrere gleichzeitige logische Übertragungsleitungen zwischen Endpunkten.
- **WS-Manager:** Unterstützt Anwendungen über die Steuerung der Nachrichtenübertragung und die Steuerung von Warteschlangen.

In Tabelle 16 ist aufgeführt, welche Komponenten für welche Verbindungsart erforderlich sind. Bei der Client/Server-Verbindungsart kann MQSeries Everyplace im Client- oder im Servermodus betrieben werden. Bei der Servlet-Option wird MQSeries Everyplace als HTTP-Servlet konfiguriert, wobei der HTTP-Server selbst für eingehende Verbindungsanforderungen empfangsbereit sein muss.

Tabelle 16. Verbindungsarten

	WS-Manager	Kanalmanager	Kanalempfangsprogramm
Client	Ja		
Client/Server	Ja	Ja	Ja
Peer	Ja		
Server	Ja	Ja	Ja
Servlet	Ja	Ja	

Welche Verbindungsart von den WS-Managern verwendet wird, hat keine direkten Auswirkungen auf die MQSeries Everyplace-Anwendungen. Die Verbindungsart legt jedoch fest, welche Ressourcen für die Partner zur Verfügung stehen, zwischen welchen WS-Managern eine Verbindung hergestellt werden kann, wie groß der Speicherbedarf für MQSeries Everyplace ist und welche Verbindungen gleichzeitig bestehen können.

Peer-zu-Peer-Verbindung

Peer-zu-Peer-Kanäle weisen die Möglichkeiten auf, die einem einzelnen Kanal mit einem Kanalmanager und einem Kanalempfangsprogramm zur Verfügung stehen. Bei der Erstellung eines Peer-zu-Peer-Kanals zwischen zwei

Verbindungsarten

WS-Managern übernimmt einer von diesen die Aufgaben des Empfangsprogramms, während der andere WS-Manager Verbindungsanforderungen absetzt. Ein WS-Manager, der für diese Verbindungsart konfiguriert ist, kann mehrere Peer-zu-Peer-Verbindungen mit anderen WS-Managern einleiten, jedoch nur jeweils auf eine eingehende Verbindungsanforderung reagieren; weitere Verbindungsanforderungen können immer erst nach Beendigung des vorherigen Peer-zu-Peer-Kanals bearbeitet werden. Die beiden beteiligten WS-Manager können über jeden beliebigen Peer-zu-Peer-Kanal Vorgänge einleiten; so können beispielsweise Anwendungen auf den WS-Managern jeweils auf die Warteschlangen des anderen WS-Managers zugreifen.

Peer-zu-Peer-Kanäle können unter Umständen nicht bei Vorhandensein einer Firewall verwendet werden, da das Ziel der eingehenden Verbindungsanforderung eventuell von der Firewall zurückgewiesen wird.

Client/Server-Verbindung

Standardkanäle für Client/Server-Verbindungen verfügen nicht über die Möglichkeiten, Verbindungsanforderungen zu erkennen, daher ist ein unabhängiges Empfangsprogramm auf dem Server erforderlich; darüber hinaus muss auf dem Server ein Kanalmanager für die Steuerung mehrerer gleichzeitiger Kanäle vorhanden sein. Der Client setzt eine Verbindungsanforderung ab, auf die der Server reagiert. Ein Server kann in der Regel gleichzeitige Anforderungen mehrerer Clients verarbeiten. Über einen Standardkanal erhält der Client Zugriff auf die Serverressourcen. Ist für eine Anwendung auf dem Server der synchrone Zugriff auf die Client-Ressourcen erforderlich, muss ein zweiter Kanal erstellt werden, bei dem die Rollen vertauscht sind. Da Standardkanäle jedoch bidirektional sind, werden die an einen Client gerichteten Nachrichten aus der Übertragungswarteschlange des Servers über den Standardkanal (Client/Server-Kanal) zugestellt, der vom Client eingeleitet wurde.

Ein Client kann mehreren Servern gleichzeitig zugeordnet sein. Für diese Konfiguration ist kein Kanalmanager erforderlich, da Kanalmanager mehrere eingehende Kanäle handhaben können.

Client/Server-Verbindungen können im Allgemeinen bei Vorhandensein von Firewalls eingesetzt werden, da das Ziel der eingehenden Verbindung in der Regel von der Firewall zugelassen wird.

Mehrere Verbindungsarten

Ein einzelner WS-Manager kann Peer-zu-Peer- oder Client/Server-Verbindungen einleiten und entweder als Server oder als Peer reagieren. In diesem Fall müssen dem Peer-Kanalempfangsprogramm und dem Standardkanalempfangsprogramm verschiedene Anschlussnummern zugeordnet sein.

Klassen

MQSeries Everyplace stellt für verschiedene Funktionen eine Auswahl von Klassen zur Verfügung, mit denen der MQSeries Everyplace-Betrieb an die spezifischen Anwendungsanforderungen angepasst werden kann. In einigen Fällen werden die Schnittstellen zu Klassen dokumentiert, so dass die Entwicklung zusätzlicher Lösungen ermöglicht wird. Tabelle 17 enthält eine Übersicht über diese Möglichkeiten. Klassen können entweder direkt oder über Aliasnamen identifiziert werden.

Tabelle 17. Klassenoptionen

Klasse	Verfügbare Alternative	Dokumentierte Schnittstellen
Verwaltung	Nein	Ja
Authentifizierungsfunktionen	Ja	Nein
Kommunikationsadapter	Ja	Ja
Übertragungsarten	Ja	Nein
Komprimierungsfunktionen	Ja	Nein
Verschlüsselungsfunktionen	Ja	Nein
Ereignisprotokoll	Beispiel verfügbar	Ja
Nachrichten	Nein	Ja
Warteschlangenspeicher	Ja	Nein
Regeln	Standardklassen verfügbar	Ja
Trace	Beispiele verfügbar	Ja

Anwendungen laden

Wird ein MQSeries Everyplace-WS-Manager als Client (oder Peer) konfiguriert, ist die Anwendung, von der die Anforderung abgesetzt wird, für das Laden anderer Anwendungen in die JVM verantwortlich. Hierzu können entweder standardmäßige Java-Funktionen verwendet werden, oder das Klassenladeprogramm, das Bestandteil von MQSeries Everyplace ist, muss verfügbar sein. Dies ermöglicht die gleichzeitige Ausführung mehrerer Anwendungen für einen einzelnen WS-Manager auf derselben JVM. Ebenso können mehrere JVMs verwendet werden, allerdings ist für jede ein eigener WS-Manager und ein eindeutiger Name erforderlich.

Wird ein MQSeries Everyplace-WS-Manager als Server konfiguriert, ist MQSeries Everyplace selbst die Anwendung, von der Verbindungsanforderungen abgesetzt werden. MQSeries Everyplace unterstützt das vorherige Laden einer Klassenliste; die in dieser Liste aufgeführten Klassen werden zuerst geladen, dann erst der WS-Manager selbst.

Kapitel 6. MQSeries Everyplace- und MQSeries-Netze

Ein MQSeries Everyplace-Netz kann zwar als Standalone-Installation, d. h. ohne MQSeries-Server oder -Netz betrieben werden, wird aber häufig als Ergänzung zu einer vorhandenen MQSeries-Installation eingesetzt. Diese kann dadurch um neue Plattformen und Einheiten oder erweiterte Leistungsmerkmale wie beispielsweise warteschlangen- oder nachrichtenbasierte Sicherheit und synchrone Nachrichtenübertragung erweitert werden. Für eine MQSeries Everyplace-Anwendung stellen die MQSeries-Warteschlangen und -WS-Manager lediglich zusätzliche ferne Warteschlangen bzw. WS-Manager dar. Allerdings bestehen hier einige Einschränkungen, da der Zugriff auf diese Warteschlangen nicht direkt über dynamische MQSeries Everyplace-Kanäle und einen MQSeries Everyplace-WS-Manager erfolgen kann, sondern ein MQSeries Everyplace-Gateway erforderlich ist. Das Gateway kann über MQSeries-Client-Kanäle direkt oder indirekt Nachrichten an mehrere MQSeries-WS-Manager senden. Bei einer indirekten Verbindung werden die Nachrichten über MQSeries-Client-Kanäle an zwischengeschaltete MQSeries-WS-Manager gesendet und von dort über MQSeries-Nachrichtenkanäle an den Ziel-WS-Manager übertragen.

Nachrichten von einer MQSeries-Anwendung an MQSeries Everyplace werden wie üblich an den MQSeries Everyplace-WS-Manager und die MQSeries Everyplace-Warteschlange gerichtet. Hierzu wird das MQSeries-Routing (die Definition des fernen WS-Managers) so definiert, dass die MQSeries Everyplace-Nachrichten in besonderen MQSeries-Übertragungswarteschlangen eingereiht werden. MQSeries-Kanäle sind nicht wie sonst für die Übertragungswarteschlangen definiert. Statt dessen ruft das MQSeries Everyplace-Gateway die Nachrichten aus den Warteschlangen ab und sorgt dafür, dass sie dem MQSeries Everyplace-Ziel zugestellt werden. Die Anzahl der Übertragungswarteschlangen (d. h. der Übertragungswege) ist konfigurierbar und sollte dem Nachrichtenvolumen entsprechen.

Schnittstellen zu MQSeries

Die MQSeries Everyplace-Architektur unterstützt den Einsatz einer oder mehrerer optionaler Brückentypen zwischen MQSeries Everyplace und anderen Nachrichtenübertragungssystemen.

In MQSeries Everyplace Version 1.2 wird nur eine Brücke, die *MQSeries-Brücke*, unterstützt, die als Schnittstelle zwischen MQSeries Everyplace- und MQSeries-Netzen fungiert. Diese Brücke stellt über den MQSeries-Java-Client eine Verbindung zu einem oder mehreren MQSeries-WS-Managern her und ermöglicht damit den Austausch von Nachrichten zwischen MQSeries Everyplace und MQSeries. In der aktuellen Version von MQSeries Everyplace wird pro Server nur der Einsatz einer Brücke empfohlen; jede dieser Brücken ist mit mehreren *MQSeries-Proxy-WS-Managern* (den Definitionen von MQSeries-WS-Managern) verbunden. Für jeden MQSeries-WS-Manager, der mit MQSeries Everyplace kommuniziert, ist die Definition eines Proxy-WS-Managers erforderlich. Jeder dieser Definitionen können ein oder mehrere *Client-Verbindungsservices* zugeordnet werden, wobei jeder die Verbindung zu einem einzelnen MQSeries-WS-Manager darstellt. Diese Client-Verbindungsservices können jeweils unterschiedliche MQSeries-Serververbindungen zum WS-Manager sowie unterschiedliche Merkmale wie beispielsweise Benutzer-Exits oder Anschlüsse verwenden.

Ein Gateway kann über eine Reihe von *Empfangsprogrammen* verfügen, über die es eine Verbindung zum MQSeries-WS-Manager herstellen und Nachrichten aus MQSeries in MQSeries Everyplace abrufen kann. Ein Empfangsprogramm verwendet nur jeweils einen Service, um diese Verbindung herzustellen, und jedes Empfangsprogramm stellt nur eine Verbindung zu jeweils einer Übertragungswarteschlange auf dem MQSeries-WS-Manager her. Jedes Empfangsprogramm überträgt die Nachrichten aus einer MQSeries-Übertragungswarteschlange über den übergeordneten Gateway-WS-Manager an ein beliebiges Ziel im MQSeries Everyplace-Netz. Damit kann ein einzelner Gateway-WS-Manager Nachrichten aus mehreren MQSeries-Nachrichtenquellen in das MQSeries Everyplace-Netz leiten.

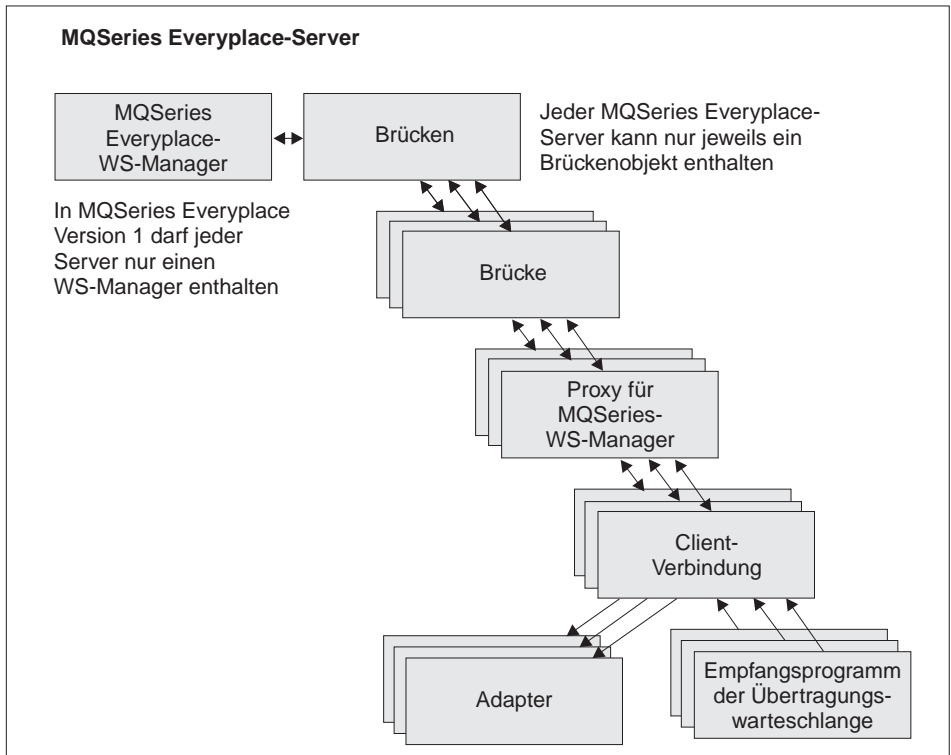


Abbildung 9. MQSeries-Brückenobjekthierarchie

Bei der Nachrichtenübertragung von MQSeries Everyplace an MQSeries konfiguriert der Gateway-WS-Manager ein oder mehrere *Brückenwarteschlangenobjekte*. Jedes dieser Objekte kann direkt eine Verbindung zu einem beliebigen WS-Manager herstellen und die Nachrichten an die Zielwarteschlangen senden. Auf diese Weise kann ein Gateway die über einen einzelnen MQSeries Everyplace-WS-Manager weitergeleiteten MQSeries Everyplace-Nachrichten direkt oder indirekt an einen beliebigen MQSeries-WS-Manager zustellen. In Tabelle 18 sind die Merkmale des Brückenobjekts aufgeführt.

Tabelle 18. Merkmale des Brückenobjekts

Merkmal	Beschreibung
Brückenname	Eine Liste mit Brücken
Betriebsstatus	Der Status ist aktiv oder gestoppt

Das Brückenobjekt und die anderen Gateway-Objekte können unabhängig vom MQSeries Everyplace-WS-Manager gestartet und gestoppt werden. Wird ein Gateway-Objekt gestartet bzw. gestoppt, werden alle untergeordneten Objekte (d. h. alle Brücken, Proxy-WS-Manager, Client-Verbindungen und

Verbindung mit MQSeries

Empfangsprogramme der Übertragungswarteschlangen) ebenfalls gestartet bzw. gestoppt. In Tabelle 19 sind die Brückenmerkmale aufgeführt.

Tabelle 19. Brückenmerkmale

Merkmal	Beschreibung
Klasse	Brückenklasse
Standardumsetzungsprogramm	Die Standardklasse (Regelklasse) für die Umsetzung von MQSeries Everyplace- in MQSeries-Nachrichten oder umgekehrt; diese Klasse kommt zur Anwendung, wenn der Zielwarteschlange keine andere Umsetzungsprogrammklasse zugeordnet ist.
Intervall der Überwachungssignale	Die Basiszeitgebereinheit für die Ausführung von Aktionen für Brückenobjekte
Name	Der Name des Brückenobjekts
Betriebsstatus	Der Status ist aktiv oder gestoppt
Startregelklasse	Die Regelklasse, die beim Start des Brückenobjekts verwendet wird
Untergeordnete MQSeries-Proxy-WS-Manager	Eine Liste mit allen Proxy-WS-Managern, die der Brücke zugeordnet sind

Für einfache Fälle kann die gesamte Nachrichtenumsetzung von einem Standardumsetzungsprogramm (Regel) ausgeführt werden. Diese Einstellung kann geändert werden, indem zusätzlich ein Umsetzungsprogramm pro Empfangsprogramm (für Nachrichten von MQSeries an MQSeries Everyplace) konfiguriert wird. Für eine erweiterte Steuerung können die Umsetzungsregeln über die Definitionen von Brückenwarteschlangen auf dem Gateway für jede Zielwarteschlange konfiguriert werden. Dies gilt für die MQSeries Everyplace- und für die MQSeries-Zielwarteschlangen.

Der MQSeries-Proxy-WS-Manager beschreibt die Merkmale eines bestimmten MQSeries-WS-Managers. In Tabelle 20 sind die Proxy-Merkmale aufgeführt.

Tabelle 20. MQSeries-Proxy-WS-Manager - Merkmale

Merkmal	Beschreibung
Klasse	Die Klasse "MQSeries-Proxy-WS-Manager"
Name des MQSeries-Host	IP-Host-Name, mit dem über die Java-Client-Klassen die Verbindungen zum MQSeries-WS-Manager hergestellt werden. Falls keine Angabe erfolgt, wird davon ausgegangen, dass sich der MQSeries-WS-Manager auf derselben Maschine wie die Brücke und die verwendeten Java-Bindungen befindet.

Tabelle 20. MQSeries-Proxy-WS-Manager - Merkmale (Forts.)

Merkmal	Beschreibung
Name des MQSeries-Proxy-WS-Managers	Der Name des MQSeries-WS-Managers
Name der zugeordneten Brücke	Der Name des Brückenobjekts, dem dieser MQSeries-Proxy-WS-Manager zugeordnet ist
Betriebsstatus	Der Status ist aktiv oder gestoppt
Startregelklasse	Die Regelklasse, die beim Start des MQSeries-WS-Managerobjekts verwendet wird
Untergeordnete Client-Verbindungen	Eine Liste mit allen Client-Verbindungsobjekten, die diesem Proxy zugeordnet sind

Die Definition des Gateway-Verbindungsservices enthält alle Angaben, die für die Herstellung einer Verbindung zu einem MQSeries-WS-Manager erforderlich sind. In Tabelle 21 sind die Verbindungsmerkmale aufgeführt.

Tabelle 21. Client-Verbindungsservice - Merkmale

Merkmal	Beschreibung
Adapterklasse	Die Klasse, die als Gateway-Adapter verwendet wird
CCSID*	Die ganze Zahl, die als MQSeries-CCSID verwendet wird
Klasse	Die Klasse "Brücken-Client-Verbindungsservice"
Maximale Verbindungsleerlaufzeit	Gibt die maximal zulässige Leerlaufzeit für eine Verbindung an (d. h. die Zeit, in der die Verbindung nicht genutzt wird), bevor sie beendet wird.
MQSeries-Kennwort*	Das vom Java-Client verwendete Kennwort
MQSeries-Anschluss*	IP-Anschlussnummer, die für die Herstellung von Verbindungen zum MQSeries-WS-Manager über die Java-Client-Klassen verwendet wird. Falls keine Angabe erfolgt, wird davon ausgegangen, dass sich der MQSeries-WS-Manager auf derselben Maschine wie die Brücke und die verwendeten Java-Bindungen befindet.
MQSeries-Empfangs-Exit-Klasse*	Wird zum Abgleich mit dem Empfangs-Exit am anderen Ende des Client-Kanals verwendet; dem Exit ist eine Zeichenfolge zugeordnet, die die Übergabe von Daten an den Exit-Code ermöglicht.
MQSeries-Sicherheits-Exit-Klasse*	Wird zum Abgleich mit dem Sicherheits-Exit am anderen Ende des Client-Kanals verwendet; dem Exit ist eine Zeichenfolge zugeordnet, die die Übergabe von Daten an den Exit-Code ermöglicht.

Verbindung mit MQSeries

Tabelle 21. Client-Verbindungsservice - Merkmale (Forts.)

Merkmak	Beschreibung
MQSeries-Sende-Exit-Klasse*	Wird zum Abgleich mit dem Sende-Exit am anderen Ende des Client-Kanals verwendet; dem Exit ist eine Zeichenfolge zugeordnet, die die Übergabe von Daten an den Exit-Code ermöglicht.
MQSeries-Benutzer-ID*	Die vom Java-Client verwendete Benutzer-ID
Der Name des Client-Verbindungsservices	Der Name des Serververbindungskanals auf der MQSeries-Maschine
Name des zugehörigen Proxy-WS-Managers	Der Name des zugehörigen Proxy-WS-Managers
Startregelklasse	Die Regelklasse, die beim Start des Objekts "Brücken-Client-Verbindungsservice" verwendet wird
Name der Sync-Warteschlange	Der Name der MQSeries-Warteschlange, die von der Brücke für Synchronisationszwecke verwendet wird
Löschregelklasse für Sync-Warteschlange	Die zu verwendende Regelklasse, wenn in der Sync-Warteschlange eine Nachricht vorgefunden wird
Betriebsstatus	Der Status ist aktiv oder gestoppt
Name der Brücke	Der Name des Brückenobjekts, dem diese Client-Verbindung zugeordnet ist
Untergeordnete Empfangsprogramme der MQ-XMIT-Warteschlange	Eine Liste mit allen Empfangsprogramobjekten, die diese Client-Verbindung verwenden
*Ausführliche Informationen zu diesen Parametern finden Sie im Handbuch <i>MQSeries Using Java</i>	

Die *Adapterklasse* wird zur Nachrichtenübertragung von MQSeries Everyplace an MQSeries, die *Sync-Warteschlange* zur Statusüberwachung des Übertragungsvorgangs verwendet. Der Inhalt dieser Warteschlange wird bei Wiederherstellungen verwendet, um so eine gesicherte Nachrichtenübertragung zu gewährleisten; nach einem normalen Systemabschluss ist diese Warteschlange leer. Diese Klasse kann über mehrere Client-Verbindungen und für mehrere Brückendefinitionen verwendet werden, vorausgesetzt, es handelt sich jeweils um dieselben Empfangs-, Sende- und Sicherheits-Exits. Darüber hinaus können in dieser Warteschlange abhängig von den bestehenden Empfangsprogrammerkmalen Statusinformationen zu Nachrichtenübertragungen von MQSeries an MQSeries Everyplace gespeichert werden. Die *Löschregelklasse für die Sync-Warteschlange* kommt zur Anwendung, wenn in der Sync-Warteschlange eine Nachricht vorgefunden wird. Dies deutet darauf hin, dass eine Nachricht von MQSeries Everyplace nicht bestätigt wurde.

Die maximale Verbindungsleerlaufzeit wird zur Steuerung des Pools mit Java-Client-Verbindungen verwendet, der vom Brücken-Client-Verbindungsservice zum MQSeries-System verwaltet wird. Wird eine MQSeries-Verbindung nicht genutzt, wird ein Zeitgeber gestartet, bei dessen Ablauf die Verbindung beendet wird, falls sie in der Zwischenzeit nicht genutzt wird. Da die Herstellung von MQSeries-Verbindungen äußerst aufwändig ist, stellt dieser Vorgang eine effiziente Wiederverwendung der Verbindungen ohne eine übermäßige Ressourcenbelegung sicher. Der Wert "Null" gibt an, dass kein Verbindungs-Pool verwendet werden soll.

Die Merkmale des Empfangsprogramms, das Nachrichten von MQSeries an MQSeries Everyplace überträgt, sind in Tabelle 22 aufgeführt.

Tabelle 22. Empfangsprogramm - Merkmale

Merkmal	Beschreibung
Klasse	Die Empfangsprogrammklass
Name der Warteschlange für nicht zugestellte Nachrichten	Die Warteschlange, in die Nachrichten von MQSeries an MQSeries Everyplace eingereiht werden, die nicht zugestellt werden können
Adapter des Empfangsprogramms für Statusinformationen	Der Klassenname des Adapters, in dem Statusinformationen gespeichert werden
Empfangsprogrammname	Der Name der MQSeries-XMIT-Warteschlange, die Nachrichten zur Verfügung stellt
Name des zugeordneten Client-Verbindungsservices	Der Name des Client-Verbindungsservices
Betriebsstatus	Der Status ist aktiv oder gestoppt
Startregelklasse	Die Regelklasse, die beim Start des Empfangsprogramms verwendet wird
Umsetzungsprogrammklasse	Die Regelklasse, die bei der Umsetzung von MQSeries- in MQSeries Everyplace-Nachrichten verwendet wird
Regelklasse für nicht zugestellte Nachrichten	Die Regelklasse, die die entsprechende Maßnahme für den Fall, dass Nachrichten von MQSeries an MQSeries Everyplace nicht zugestellt werden können, festlegt
Wartezeit auf Nachrichten in Sekunden	Eine erweiterte Option, mit der die Leistung des Empfangsprogramms in Ausnahmesituationen gesteuert werden kann

Verbindung mit MQSeries

Die *Regelklasse für nicht zugestellte Nachrichten* legt die entsprechende Maßnahme für den Fall, dass eine Nachricht von MQSeries an MQSeries Everyplace nicht zugestellt werden kann, fest. In der Regel werden solche Nachrichten in die *Warteschlange für nicht zustellbare Nachrichten* des MQSeries-Systems eingereiht.

Um eine sichere Nachrichtenzustellung zu ermöglichen, verwendet die Empfangsprogrammmklasse den *Adapter des Empfangsprogramms für Statusinformationen* zum Speichern von Statusinformationen. Diese werden entweder im MQSeries Everyplace-System oder in der Sync-Warteschlange des MQSeries-Systems gespeichert.

Das Empfangsprogramm der Übertragungswarteschlange ermöglicht fernen MQSeries-Warteschlangen den Verweis auf lokale MQSeries Everyplace-Warteschlangen. Darüber hinaus können Sie auch ferne MQSeries Everyplace-Warteschlangen erstellen, die auf lokale MQSeries-Warteschlangen verweisen. Diese Definitionen ferner MQSeries Everyplace-Warteschlangen werden als *MQSeries-Brückenwarteschlangen* bezeichnet; über sie können Nachrichten in eine MQSeries-Warteschlange eingereiht, aus einer MQSeries-Warteschlange abgerufen bzw. die in einer MQSeries-Warteschlange enthaltenen Nachrichten durchsucht werden.

Die Definition einer MQSeries-Brückenwarteschlange kann folgende Attribute enthalten:

Tabelle 23. MQSeries-Brückenwarteschlange - Merkmale

Merkmal	Beschreibung
Aliasname	Alternative Namen für die Warteschlange
Authentifizierungsfunktion	Muss auf Null gesetzt werden
Klasse	Die Objektklasse
Client-Verbindung	Der Name des Client-Verbindungsservices, der verwendet werden soll
Komprimierungsfunktion	Muss auf Null gesetzt werden
Verschlüsselungsfunktion	Muss auf Null gesetzt werden
Zeitlimitüberschreitung	Wird an das Umsetzungsprogramm übergeben
Maximal zulässige Nachrichtenlänge	Wird an die Regelklassen übergeben
Modus	Muss auf "synchron" gesetzt werden
MQ-Proxy-WS-Manager	Name des MQSeries-WS-Managers, an den die Nachricht zuerst gesendet werden soll
MQSeries-Brücke	Der Name der Brücke, über die die Nachricht an MQSeries übertragen werden soll

Table 23. MQSeries-Brückenwarteschlange - Merkmale (Forts.)

Merkm ^{al}	Beschreibung
Name	Der Name, unter dem die ferne MQSeries-Warteschlange in MQSeries Everyplace bekannt ist
Zugeordneter WS-Manager	Name des WS-Managers, dem die Definition zugeordnet ist
Priorität	Die Priorität der Nachrichten (wird durch Angabe eines Nachrichtenwertes außer Kraft gesetzt)
Name der fernen MQSeries-Warteschlange	Der Name der fernen MQSeries-Warteschlange
Regel	Die Regelklasse, die für Warteschlangenvorgänge verwendet werden soll
Ziel-WS-Manager	Der MQSeries-WS-Manager, dem die Warteschlange zugeordnet ist
Umsetzungsprogramm	Name der Umsetzungsprogrammklasse, die Nachrichten vom MQSeries Everyplace- in das MQSeries-Format umsetzt
Art	Die MQSeries-Brücken-Warteschlange

Anmerkung: Die *Verschlüsselungsfunktions-*, *Authentifizierungsfunktions-* und *Komprimierungsfunktions-*klassen definieren eine Reihe von Warteschlangenattributen, die die Sicherheitsstufe für die Nachrichten festlegen, die an diese Warteschlange übertragen werden. Zwischen dem Zeitpunkt, zu dem eine Nachricht in MQSeries Everyplace gesendet wird, bis zu ihrer Übergabe an die MQSeries-Brückenwarteschlange gilt für diese Nachricht zumindest die für die Warteschlange definierte Sicherheitsstufe. Diese Sicherheitsstufen gelten *nicht*, wenn die Nachricht von der MQSeries-Brückenwarteschlange an das MQSeries-System gesendet wird; für diese Übertragung wird die Sicherheit der Send- und Empfangs-Exits der Client-Verbindung verwendet. Es erfolgt keine Überprüfung, ob die Sicherheitsstufe der Warteschlange aufrechterhalten bleibt.

MQSeries-Brückenwarteschlangen arbeiten nur im synchronen Modus. Asynchrone Anwendungen müssen daher entweder eine Kombination aus MQSeries Everyplace-Warteschlangen zum Speichern und Weiterleiten und Ausgangsserverwarteschlangen verwenden, oder sich zwischengeschalteter Definitionen asynchroner ferner Warteschlangen bedienen, wenn Nachrichten an MQSeries-Brückenwarteschlangen gesendet werden.

Verbindung mit MQSeries

Anwendungen verwenden MQSeries-Brückenwarteschlangen auf dieselbe Weise wie eine ferne MQSeries Everyplace-Warteschlange, d. h., sie verwenden die Methoden *putMessage*, *browseMessages* und *getMessage* der Klasse *MQeQueueManager*. Für den Parameter 'Warteschlangenname' in diesen Aufrufen wird der Name der MQSeries-Brückenwarteschlange angegeben, für den Parameter 'Name des WS-Managers' wird der Name des MQSeries-WS-Managers angegeben. Damit dieser WS-Manager vom lokalen MQSeries Everyplace-Server akzeptiert wird, muss jedoch für diesen MQSeries-WS-Manager eine Verbindungsdefinition vorhanden sein, in der für alle Parameter (auch für den Kanalnamen) der Wert Null angegeben ist.

Anmerkung: Die Ausführung der Methoden *getMessage* und *browseMessages* in Zusammenhang mit MQSeries-Brückenwarteschlangen ist jedoch gewissen Einschränkungen unterworfen. So können beispielsweise aus MQSeries-Brückenwarteschlangen keine Nachrichten abgerufen bzw. in ihnen durchsucht werden, die auf Definitionen ferner MQSeries-Warteschlangen verweisen. Ebenso können beim Abrufen von Nachrichten aus MQSeries-Brückenwarteschlangen keine Bestätigungs-IDs angegeben werden, die einen anderen Wert als Null haben. Dies bedeutet, dass die Methode *getMessage* für MQSeries-Brückenwarteschlangen keine gesicherte Zustellung gewährleisten kann. Für einen gesicherten Get-Vorgang müssen Sie Empfangsprogramme für Übertragungswarteschlangen verwenden, um Nachrichten aus MQSeries abzurufen.

Die Verwaltung der MQSeries-Brücke erfolgt auf dieselbe Weise wie die Verwaltung eines normalen MQSeries Everyplace-WS-Managers, d. h. über Verwaltungsnachrichten. Bei Bedarf werden für das verwaltete Objekt neue Nachrichtenklassen definiert. In Tabelle 13 auf Seite 44 sind die Verwaltungsnachrichtenklassen für das Gateway aufgeführt.

Nachrichtenumsetzung

Die an MQSeries gerichteten MQSeries Everyplace-Nachrichten werden über die Brücke gesendet und mit Hilfe eines Standardumsetzungsprogramms bzw. des Umsetzungsprogramms der Zielwarteschlangen in ein MQSeries-Format umgewandelt. Benutzerdefinierte Umsetzungsprogramme sind sehr flexibel; beispielsweise ist es sinnvoll, eine Unterklasse der MQSeries Everyplace-Nachrichtenobjektklasse für die Darstellung einer bestimmten Nachrichtenart im gesamten MQSeries Everyplace-Netz zu verwenden. Auf dem Gateway kann die Nachricht von einem Umsetzungsprogramm unter Zuordnung der entsprechenden Felder und MQSeries-Werte in ein MQSeries-Format umgesetzt werden, wobei entsprechenden Daten hinzugefügt werden, die die Bedeutung der Unterklasse angeben.

Das Standardprogramm für die Umsetzung vom MQSeries Everyplace- in das MQSeries-Format kann diese Unterklasseninformationen zwar nicht verwenden, ist jedoch trotzdem vielseitig einsetzbar. Es weist folgende Merkmale auf:

- **Nachrichtenfluss von MQSeries Everyplace an MQSeries:**

Das Standardumsetzungsprogramm für die Umsetzung vom MQSeries Everyplace- in ein MQSeries-Format arbeitet mit der Klasse **MQeMQMsgObject** zusammen. Diese Klasse stellt alle Felder dar, die möglicherweise im MQSeries-Nachrichten-Header vorhanden sind. Mit Hilfe dieser Klasse kann Ihre Anwendung unter Verwendung der Methode **set()** die erforderlichen Werte, wie beispielsweise die Priorität, setzen. Wird ein **MQeMQMsgObject** (bzw. ein von dieser Klasse abgeleitetes Objekt) an das MQSeries Everyplace-Standardumsetzungsprogramm übergeben, ruft dieses (**MQeBaseTransformer**) die Werte aus **MQeMSMsgObject** ab und setzt die entsprechenden Werte in der MQSeries-Nachricht (beispielsweise wird der Wert für die Priorität in die MQSeries-Nachricht kopiert).

Handelt es sich bei der Nachricht weder um ein **MQeMQMsgObject** noch um ein von **MQeMSMsgObject** abgeleitetes Objekt, wird die gesamte MQSeries Everyplace-Nachricht in den Hauptteil der MQSeries-Nachricht kopiert (*funneled*). Der im Nachrichtenformatfeld des MQSeries-Nachrichten-Headers gesetzte Wert gibt an, dass die MQSeries-Nachricht eine Nachricht mit diesem speziellen MQSeries Everyplace-Format enthält.

- **Nachrichtenfluss von MQSeries an MQSeries Everyplace:**

Die Umsetzung der an MQSeries Everyplace gerichteten MQSeries-Nachrichten erfolgt in ähnlicher Weise wie die Umsetzung in umgekehrter Richtung. Das Standardumsetzungsprogramm überprüft das Nachrichtenartfeld im MQSeries-Nachrichten-Header und führt die entsprechende Umsetzung durch.

Enthält der MQSeries-Nachrichten-Header den Hinweis, dass es sich um eine MQSeries Everyplace-Nachricht im **funneled**-Format handelt, wird der MQSeries-Nachrichtenhauptteil wieder in die ursprüngliche MQSeries Everyplace-Nachricht umgesetzt und an das MQSeries Everyplace-Netz gesendet.

Handelt es sich nicht um eine MQSeries Everyplace-Nachricht im **funneled**-Format, wird der Inhalt des MQSeries-Nachrichten-Headers extrahiert und in ein **MQeMQMsgObject** gestellt. Der MQSeries-Nachrichtenhauptteil wird wie ein einfaches Bytefeld gehandhabt und ebenfalls in das Objekt **MQeMQMsgObject** kopiert. Anschließend wird dieses Objekt an das MQSeries Everyplace-Netz gesendet.

Verbindung mit MQSeries

Die Verwendung der Klasse **MQeMQMsgObject** und des Standardumsetzungsprogramms ermöglicht also Folgendes:

- Eine MQSeries Everyplace-Nachricht kann unverändert über ein MQSeries-Netz an ein MQSeries Everyplace-Netz übertragen werden.
- Eine MQSeries-Nachricht kann unverändert von einem MQSeries Everyplace-Netz an ein MQSeries-Netz übertragen werden.
- Eine MQSeries Everyplace-Anwendung kann eine beliebige MQSeries-Anwendung steuern, ohne dass die MQSeries-Anwendung geändert wird.

Funktion

Ferne MQSeries-Warteschlangen ermöglichen das synchrone Einreihen (Put) von MQSeries Everyplace-Nachrichten von einem MQSeries Everyplace-WS-Manager aus; alle anderen Vorgänge im Zusammenhang mit der Nachrichtenübertragung müssen asynchron erfolgen.

MQSeries Everyplace-Verwaltungsnachrichten können nicht an einen MQSeries-WS-Manager gesendet werden, da dieser über keine Verwaltungswarteschlangen verfügt und sich das Format der Verwaltungsnachrichten vom MQSeries-Format unterscheidet.

Kompatibilität

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

Adressierung und Benennung:

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

Anwendungen:

MQSeries Everyplace unterstützt vorhandene MQSeries-Anwendungen, ohne dass hierzu Änderungen erforderlich sind.

Kanäle:

MQSeries Everyplace-Gateways verwenden MQSeries-Client-Kanäle.

Nachrichtenaustausch und -inhalt:

- Nachrichtenaustausch zwischen MQSeries Everyplace und MQSeries
- Transparentes Nachrichtennetz (Nachrichten können ohne Änderungen zwischen MQSeries Everyplace- und MQSeries-Netzen ausgetauscht werden)
- Gegenseitige Unterstützung gekennzeichnete Felder im MQSeries-Nachrichten-Header
- Einmalige gesicherte Nachrichtenzustellung

MQSeries Everyplace Version 1.2 unterstützt nicht alle MQSeries-Funktionen. Einige der wichtigsten Unterschiede (neben umgebungs-, betriebssystem- und übertragungsspezifischen Überlegungen) sind nachfolgend aufgeführt. Es sollte jedoch bedacht werden, dass in MQSeries Everyplace eine Reihe von Möglichkeiten zur Verfügung stehen, mit denen eine Vielzahl von Anwendungsfunktionen ausgeführt werden können - beispielsweise durch die Verwendung von MQSeries Everyplace-Merkmalen, die Nutzung der Unterklassenfunktion, die Verwendung eigener an Stelle der gelieferten Klassen oder durch Nutzung der Regeln, Schnittstellen und anderen Anpassungsmerkmalen, die in diesem Produkt verfügbar sind.

- Keine Clusterunterstützung
- Keine Unterstützung von Verteilerlisten
- Keine Unterstützung von Nachrichtensegmentierungen bzw. Anordnung von Nachrichten in Gruppen
- 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

Es bestehen außerdem Unterschiede in der Skalierbarkeit und den Leistungsmerkmalen.

Gesicherte Zustellung

Sowohl MQSeries Everyplace als auch MQSeries ermöglichen eine gesicherte Nachrichtenzustellung, allerdings bestehen Unterschiede in der Sicherheitsstufe. Die gesicherte Übertragung einer Nachricht von MQSeries Everyplace an MQSeries kann nur bei der gemeinsamen Verwendung von **putMessage** und **confirmPutMessage** gewährleistet werden (siehe „WS-Managerkonfiguration“ auf Seite 37). Die Nachrichtenübertragung von MQSeries an MQSeries Everyplace ist nur gesichert, wenn die MQSeries-Nachricht als permanent definiert ist.

Kapitel 7. Programmierschnittstellen

Bei der *MQSeries Everyplace-Systemprogrammierschnittstelle* handelt es sich um die Programmierschnittstelle zu MQSeries Everyplace. Zwei Programmiersprachen, Java und C, werden unterstützt:

Die Java-Version ermöglicht den Zugriff auf alle MQSeries Everyplace-Funktionen. Ausführliche Informationen zu Klassen, Methoden und Prozeduren finden Sie im Handbuch *MQSeries Everyplace for Multiplatforms Programming Reference*; Programmierbeispiele für MQSeries Everyplace finden Sie im Handbuch *MQSeries Everyplace for Multiplatforms Programming Guide*.

Die **C-Unterstützung für Palm** ermöglicht den Zugriff auf einen Teil der MQSeries Everyplace-Funktionen für Palm-Einheiten. Ausführliche Informationen zu diesen Klassen und Prozeduren sowie Programmierhinweise finden Sie im Handbuch *MQSeries Everyplace Native Client Information*.

Kapitel 8. MQSeries Everyplace - Erste Schritte

Bei MQSeries Everyplace handelt es sich um eine Produktfamilie, die Tools für die Entwicklung, den Einsatz und die Verwaltung von MQSeries Everyplace-Messaging und MQSeries Everyplace-Queuing-Lösungen zur Verfügung stellt. Die Produktfamilie umfasst Folgendes:

1. Das *Lizenzprodukt MQSeries Everyplace*, das entweder auf einem physischen Datenträger von IBM erworben oder aus dem Internet unter <http://software.boulder.ibm.com/dl/mqsem/mqsem-p> heruntergeladen werden kann). Das Lizenzprodukt umfasst folgende Komponenten:
 - MQSeries Everyplace-Java-Klassen
 - Hilfsklassen
 - Beispielquellcode für Anwendungen
 - Dienstprogramme
 - Referenzhandbücher
 - Lizenzinformationen

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

2. MQSeries Everyplace-SupportPacs (diese können aus dem Internet unter <http://software.boulder.ibm.com/dl/mqsem/mqsem-p> (wie oben) oder unter <http://www.ibm.com/software/mqseries/everyplace> heruntergeladen werden). Hierbei handelt es sich um wichtige Ergänzungen des Lizenzprodukts. Diese umfassen unter anderem folgende Komponenten:

EAP1: MQSeries Everyplace - Device code for Palm OS

Unterstützung der Programmiersprache C für die Anwendungsentwicklung in MQSeries Everyplace Version 1.0.1 unter Palm OS. (Dieser Code steht auch auf der Produkt-CD in der Datei eap1.zip zur Verfügung.)

EP01: MQSeries Everyplace - Performance Report

Analyse der MQSeries Everyplace-Leistung auf zahlreichen Client-Plattformen.

ES01: MQSeries Everyplace - Administration Tool (MQExplorer V1.0)

Ein generisches Tool für alle Java-Plattformen, das eine einfache grafische Verwaltung von MQSeries Everyplace-Warteschlangenmanagern ermöglicht.

ES02: MQSeries Everyplace - Explorer (MQe_Explorer V1.2)

Ein MQSeries Everyplace-Verwaltungs-Tool, das ausschließlich für die Unterstützung der Betriebssysteme von Microsoft Windows entwickelt wurde.

Die Verwaltungs-Tools in den MQSeries Everyplace-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.

MQSeries Everyplace verwenden

Auf Grund der unterschiedlichen Einsatzbereiche für MQSeries Everyplace unterscheidet sich dieses Produkt von anderen MQSeries-Produkten hinsichtlich Installation, Konfiguration und Einsatz. Grundsätzlich wird davon ausgegangen, dass sich der Einsatz von MQSeries Everyplace in einem Unternehmen in drei Schritten vollzieht:

1. Entwicklung und Erstellung von Prototypen

In der Lern- und Entwicklungsphase, in der auch erste Prototypen erstellt werden, kann MQSeries Everyplace kostenlos installiert und verwendet werden; dabei gelten die Bedingungen der IBM für eine MQSeries Everyplace-Entwicklungslizenz. In dieser Phase werden MQSeries Everyplace-Anwendungen mit Hilfe der MQSeries Everyplace-Java-Klassen und der MQSeries Everyplace-C-Routinen erstellt. Diese Anwendungen können auf verschiedene Weise gepackt werden:

- Beispielsweise kann ein MQSeries Everyplace-WS-Manager als Dämon eingerichtet werden; eine oder mehrere Anwendungen werden in derselben JVM (Java Virtual Machine) gestartet, die denselben WS-Manager benutzen.
- Die Anwendung enthält die erforderlichen MQSeries Everyplace-Klassen; die Anwendung läuft auf Maschinen, auf denen MQSeries Everyplace nicht installiert ist und startet einen eigenen WS-Manager in der eigenen JVM.
- Die Anwendung verwendet die auf der Zielmaschine vorhandenen MQSeries Everyplace-Klassen.

In der Entwicklungslizenz ist keine Unterstützung durch IBM enthalten. Eine Unterstützung auch über die Anwendungsentwicklung hinaus bietet die Einsatzlizenz (siehe unten).

2. Einsatzphase

In dieser Phase werden die entwickelten Anwendungen installiert, konfiguriert und eingesetzt; daher sind entsprechend den Bedingungen der MQSeries Everyplace-Lizenz von IBM Kapazitätseinheiten für die Verwendung des Produkts erforderlich. Ein Vertrieb der Klassen zusammen mit den Anwendungen ist nur in Absprache mit IBM möglich oder falls die Benutzer bereits über eine Berechtigung zur Verwendung der Klassen verfügen. Andernfalls müssen die erforderlichen Klassen von den Benutzern selbst zur Verfügung gestellt werden.

3. Verwaltungsphase

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

Auf Grund dieses Einsatzkonzeptes werden auch Unterschiede bei der Unterstützung gemacht. Bei MQSeries Everyplace mit Kapazitätseinheiten (und SupportPacs der Kategorie 3) unterscheidet IBM zwischen den folgenden Plattformen:

- Plattformen, auf denen die Installation und Anwendungsentwicklung unterstützt wird:
 - Fehlerberichte zur Installation und/oder Anwendungsentwicklung sowie zum Einsatz können eingeschickt werden.
- Plattformen, auf denen der Einsatz von Anwendungen zulässig ist, jedoch nicht direkt unterstützt wird:
 - Unter Umständen sind Fehlerberichte erforderlich, damit der Fehler auf einer unterstützten Plattform reproduziert werden kann.
- Plattformen, auf denen der Einsatz von Anwendungen unterstützt wird:
 - Fehlerberichte zum Einsatz von Anwendungen können eingeschickt werden.

Einstieg

Es gibt verschiedene Möglichkeiten, sich in MQSeries Everyplace einzuarbeiten. Die Erfahrung hat jedoch gezeigt, dass die Konfiguration eines WS-Managers sowie die Einrichtung eines einfachen MQSeries Everyplace-Netzes der effizienteste Weg ist, sich mit diesem Produkt und den Konzepten, die ihm zu Grunde liegen, vertraut zu machen. Auf der Basis dieser Kenntnisse können Sie sich dann mit der Erstellung einfacher Anwendungen weiter in das Produkt einarbeiten. Allgemein hat es sich als nicht sehr sinnvoll erwiesen, sich gleich zu Beginn mit anderen Produkten der MQSeries-Produktfamilie zu beschäftigen. Später jedoch, wenn die Brückenfunktion relevant wird, sind Erfahrungen mit anderen MQSeries-Produkten unerlässlich.

Erste Schritte

Unter Berücksichtigung des eben beschriebenen Ansatzes wird neuen Benutzern empfohlen, bei der Einarbeitung in MQSeries Everyplace wie folgt vorzugehen:

Setzen Sie sich zunächst mit den im vorliegenden Handbuch (*MQSeries Everyplace for Multiplatforms Einführung*) beschriebenen grundlegenden Konzepten auseinander, und gehen Sie anschließend wie folgt vor:

1. Wenn Sie über ein System mit einem Microsoft Windows-Betriebssystem verfügen, laden Sie MQE_Explorer SupportPac ES02 (MQE_Explorer V 1.2) herunter. Die Installation des Lizenzprodukts ist nicht unbedingt erforderlich, allerdings ist ohne dieses Produkt entsprechend der Lizenz nur ein Einsatz in Entwicklungsumgebungen möglich.
2. Gehen Sie anhand der unter „Erster Einsatz von ES02: MQE_Explorer“ beschriebenen Schritte vor.

Erster Einsatz von ES02: MQE_Explorer

Zu dem MQE_Explorer gehört ein ausführliches Benutzerhandbuch, das verschiedene Wege erläutert, einen WS-Manager rasch und problemlos einzurichten. Dieses Handbuch richtet sich nicht nur an MQE_Explorer-Administratoren; so enthält es unter anderem Beispiel-Scripts, anhand derer wichtige MQSeries Everyplace-Operationen veranschaulicht werden. Das SupportPac enthält zwei ausführbare Versionen des MQE_Explorer:


MQE_ExplorerX.exe

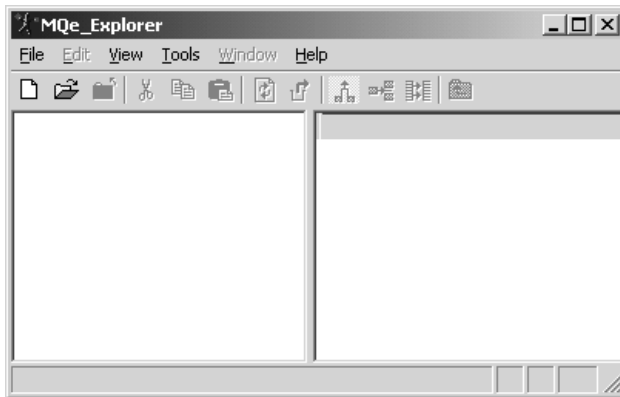
Diese Version enthält in der EXE-Datei alle für den Betrieb erforderlichen MQSeries Everyplace-Klassen. Für die Ausführung von MQE_ExplorerX.exe ist die Installation von MQSeries Everyplace nicht erforderlich. Diese Version ist besonders für Benutzer geeignet, die noch keine Erfahrungen mit dem Produkt haben.


MQE_Explorer.exe

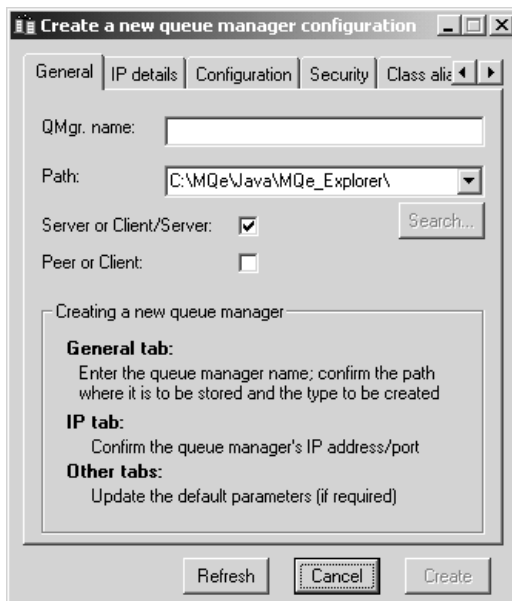
Für diese Version ist die Installation von MQSeries Everyplace Voraussetzung. Der Vorteil dieser Version ist die Verwendung der neuesten Version der MQSeries Everyplace-Bibliotheken; außerdem belegt sie wenig Speicherplatz. Diese Version ist für Entwickler und Administratoren geeignet.

Die Leistungsfähigkeit von MQSeries Everyplace soll anhand eines Beispiels veranschaulicht werden; es zeigt, wie auf einfache Weise ein WS-Manager erstellt werden kann.

1. Klicken Sie doppelt auf das Symbol für MQE_ExplorerX.exe ().
- Eine Nachricht wird angezeigt mit der Meldung, dass keine vorhandenen Komponenten gefunden wurden; klicken Sie auf **OK** (die Nachricht wird nicht mehr angezeigt). Das folgende Verwaltungsfenster wird geöffnet:

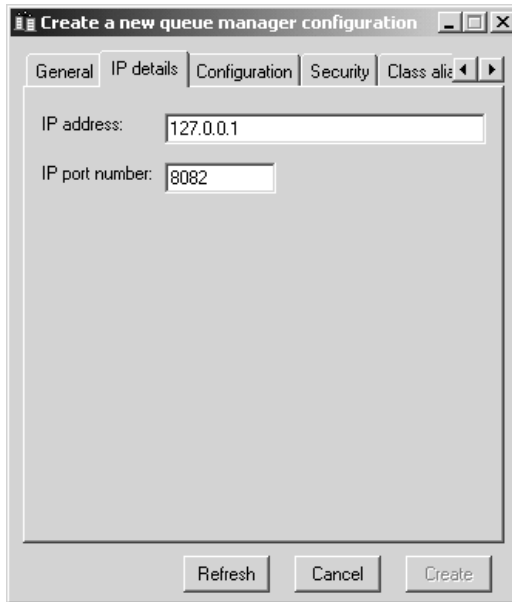


2. Klicken Sie in der Symbolleiste auf das Symbol für 'Neu' ().
Daraufhin wird ein neuer WS-Manager erstellt. Das folgende Fenster wird geöffnet:

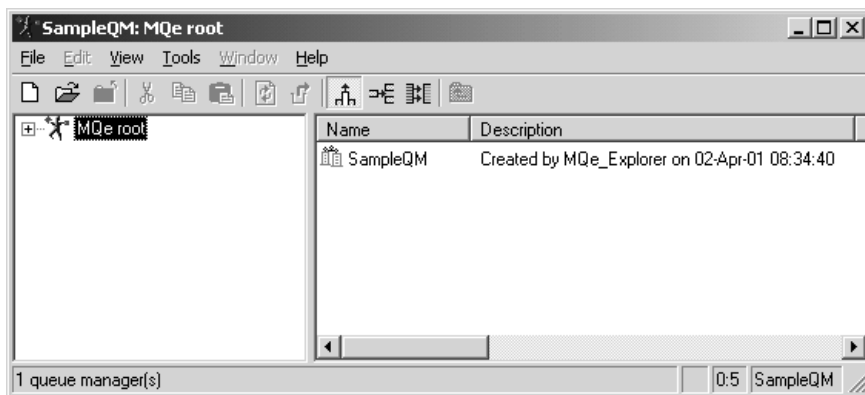


3.
 - a. Geben Sie den Namen eines WS-Managers ein (z. B. SampleQM).
 - b. Wählen Sie die Registerkarte **IP details** (IP-Angaben) aus.

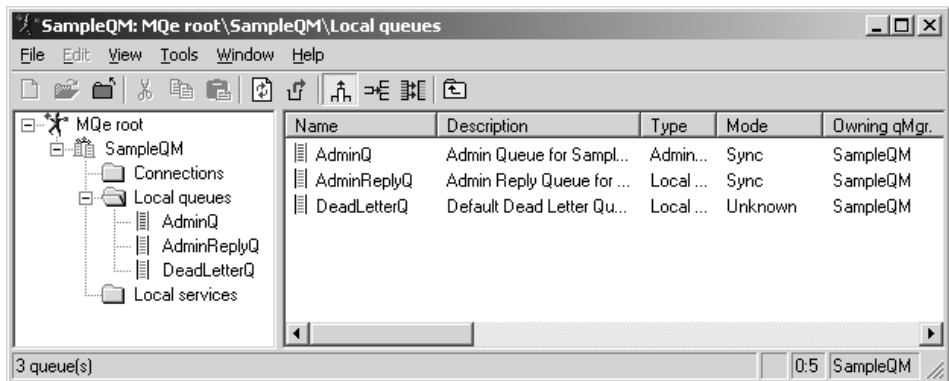
Erste Schritte



4.
 - a. Geben Sie eine IP-Adresse ein (dabei handelt es sich um die IP-Adresse der Maschine, auf der Sie arbeiten, aber es wird empfohlen, hier den Wert 127.0.0.1 (lokaler Host) einzugeben).
 - b. Klicken Sie auf **Create** (Erstellen). Daraufhin wird eine Nachricht angezeigt, die die Initialisierungsdatei angibt, die erstellt wurde (der Name dieser Datei wird für künftige Zugriffe auf diesen WS-Manager benötigt).
 - c. Klicken Sie auf **OK**; daraufhin wird das folgende Fenster geöffnet:



5. Es wurde ein WS-Manager mit Serverfunktion erstellt, der in einer eigenen JVM ausgeführt wird. Dieser WS-Manager ist am Port 8081 empfangsbereit für ankommende Client/Server-Kanalverbindungsanforderungen. Erweitern Sie in der Baumstruktur im linken Teilfenster alle Ordner, die mit einem + versehen sind; nachdem die Größe aller Fenster und Teilfenster entsprechend angepasst wurde, können Sie feststellen, dass vier Warteschlangen erstellt wurden.



6. Wenn Sie weiter üben möchten, folgen Sie den Anweisungen im Handbuch *MQ Explorer User Guide*. Dort erfahren Sie, wie Warteschlangen, Verbindungen und Nachrichten erstellt und wie MQSeries Everyplace-Netze eingerichtet werden. Sie können Anwendungen in diesen WS-Manager laden und diesen mit MQ Explorer zusammen einsetzen.

Anhang. 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 bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. An Stelle der Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen 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 dieser Veröffentlichung ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France zu richten. Anfragen an obige Adresse müssen auf englisch formuliert werden.

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) bekanntgegeben. IBM kann 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.

Bemerkungen

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung (i) den Austausch von Informationen zwischen unabhängig 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 im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Marken

Folgende Namen sind in gewissen Ländern Marken der IBM Corporation:

AIX AS/400 IBM MQSeries OS/390

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

Java sowie alle auf Java basierenden Marken und Logos sind in gewissen Ländern Marken bzw. eingetragene Marken der Sun Microsystems Inc.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken oder Dienstleistungsmarken 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, sehen Sie im Index oder im *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994, nach.

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.

Asynchrone Nachrichtenübertragung. 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.

Brücke. Ein MQSeries Everyplace-Objekt, das den Austausch von Nachrichten zwischen MQSeries Everyplace und anderen Nachrichtenübertragungssystemen, wie beispielsweise MQSeries, ermöglicht.

Client. In MQSeries eine Laufzeitkomponente, die lokalen Benutzeranwendungen Zugriff auf Services zur Steuerung von Warteschlangen auf einem Server ermöglicht.

Dynamischer Kanal. Ein dynamischer Kanal verbindet MQSeries Everyplace-Einheiten miteinander und ermöglicht die bidirektionale Übertragung synchroner und asynchroner Nachrichten bzw. Antworten.

Gateway. Bei einem MQSeries Everyplace-Gateway oder Server handelt es sich um einen Computer, auf dem neben dem MQSeries Everyplace-Code auch ein Kanalmanager installiert ist.

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

Hypertext Markup Language (HTML). Eine Sprache zur Definition von Informationen, die im World Wide Web angezeigt werden sollen.

Instanz. Ein Objekt. Es besteht die Möglichkeit, ein Objekt für eine Klasse anzulegen, das als Instanz dieser Klasse bezeichnet wird.

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.

Java Developer's Kit (JDK). Ein von Sun Microsystems für Java-Entwickler vertriebenes Softwarepaket. Es umfasst den Java-Interpreter, Java-Klassen sowie Java-Entwicklungs-Tools: Compiler, Debugger, Disassembler, Appletviewer, Stub File Generator und Dokumentations-Tool.

Java Naming and Directory Service (JNDI). Eine mit Java erstellte API. Sie stellt Namens- und Verzeichnisfunktionen für Java-Anwendungen zur Verfügung.

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

Kanalmanager. Ein MQSeries Everyplace-Objekt, das eine größere Anzahl gleichzeitiger logischer Übertragungsleitungen zwischen Endpunkten unterstützt.

Kapselung. Hierbei handelt es sich um eine objektorientierte Programmiertechnik, bei der Daten ausschließlich einem bestimmten Objekt zugeordnet werden, so dass auf die Daten nur über Methodenaufrufe zugegriffen werden kann.

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.

Lightweight Directory Access Protocol (LDAP). Ein Client/Server-Protokoll für den Zugriff auf Verzeichnisservices.

Methode. Bezeichnet in der objektorientierten Programmierung eine Funktion oder Prozedur.

MQI-Kanal. Verbindet einen MQSeries-Client mit einem WS-Manager auf einem Server und ermöglicht die bidirektionale Übertragung von MQI-Aufrufen und -Antworten.

MQSeries. Eine Gruppe von IBM Lizenzprogrammen, die Services für die Steuerung von Nachrichtenwarteschlangen zur Verfügung stellen.

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

Nachrichtenwarteschlange. Siehe *Warteschlange*.

Objekt. (1) In Java eine Instanz einer Klasse. Eine Klasse ist das Abbild einer Gruppe von Elementen, ein Objekt das Abbild eines bestimmten Elements dieser Gruppe. (2) In MQSeries ein WS-Manager, eine Warteschlange oder ein Kanal.

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.

Paket. Über Pakete wird in Java Teilen des Java-Codes der Zugriff auf bestimmte Klassengruppen ermöglicht. Damit erhält der Code-Teil in einem bestimmten Paket Zugriff auf alle Klassen in diesem Paket sowie auf alle nicht privaten Methoden und Felder dieser Klassen.

Personal Digital Assistant (PDA). Ein Taschen-PC.

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

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) Bei einem MQSeries Everyplace-Server handelt es sich um eine Einheit, auf der ein MQSeries Everyplace-Kanalmanager konfiguriert ist. (2) Bei einem MQSeries-Server handelt es sich um einen WS-Manager, der Client-Anwendungen auf fernen Workstations Services zur Steuerung von Nachrichtenwarteschlangen zur Verfügung stellt. (3) Allgemein ein Programm, das in Client/Server-Modellen auf Datenanforderungen reagiert. (4) Der Computer, auf dem ein Server-Programm zur Ausführung kommt.

Servlet. Ein Java-Programm, das nur auf einem Web-Server ausgeführt wird.

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

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*.

Transmission Control Protocol/Internet Protocol (TCP/IP). Eine Gruppe von Übertragungsprotokollen, die Peer-zu-Peer-Konnektivitätsfunktionen für lokale und Weitverkehrsnetze (LANs und WANs) unterstützen.

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

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

Warteschlange. Ein MQSeries-Objekt. Anwendungen für die Steuerung von Warteschlangen können Nachrichten in Warteschlangen stellen und aus diesen abrufen.

Web. Siehe *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 Anwendungen Services für die Steuerung von Nachrichtenwarteschlangen zur Verfügung stellt.

Literaturverzeichnis

Referenzliteratur

- *MQSeries Everyplace for Multiplatforms Wichtige Hinweise*, GC12-2861
- *MQSeries Everyplace for Multiplatforms Programming Reference*, SC34-5846
- *MQSeries Everyplace for Multiplatforms Programming Guide*, SC34-5845
- *MQSeries Everyplace for Multiplatforms Native Client Information*, GC34-5883
- *MQSeries An Introduction to Messaging and Queuing*, GC33-0805-01
- *MQSeries for Windows NT V5R1 Einstieg*, GC12-2642-00

Index

A

Adapter, MQSeries Everyplace 49
Anforderungen, kunden-
spezifische 14
Anpassung 61
Anwendungen, laden 65
Anwendungen, MQSeries Every-
place 14
Asynchrone Nachrichtenübertra-
gung 50
Attributsregeln 61
Ausgabedaten, Format 24
Ausgangsserver, MQSeries Every-
place 28
Ausgangsserver-Warteschlangen 28
Authentifizierbare Definitionen-
einheiten 57
Automatische Registrierung 59

B

Begriffe viii
Beispielkonfigurationen 11
Beschreibung 1
Betriebssysteme, unterstützte 3
Brücke, MQSeries 68
Brückenobjekt 69, 70

C

Client, MQSeries 9
Client-Kanäle 17
Client/Server-Kanäle 10
Client/Server-Verbindung 64

D

Definitionseinheiten, authentifizier-
bare 57
Dynamische Kanäle 10, 17, 47

E

Einheiten, MQSeries Everyplace 17
Empfangsprogrammobjekt 73, 74
Ereignisprotokoll 50
Erforderliche Betriebssysteme 3

F

Ferne Warteschlangen 26
Format der Ausgabedaten 24

G

Gateways, MQSeries Everyplace 17
Gesicherte Nachrichtenzu-
stellung 80

H

Hinweise, rechtliche 91
Host-Nachrichtenübertragung viii,
8

K

Kanäle 10
Kanäle, Client 17
Kanäle, dynamisch 17, 47
Kanalempfangsprogramm 63
Kanalmanager 63
Klassen, MQSeries Everyplace 65
Kompatibilität mit MQSeries 78
Komprimierung 52
Konfiguration 61
Konfigurationen, Beispiele 11
Konzepte, Produkt 17
Kundenanforderungen 14

L

Laden, Anwendungen 65
Leistungsspektrum 13
Lokale Sicherheit 53
Lokale Warteschlangen 25

M

Marken 92
Mehrere Verbindungsarten 64
Mini Certificate, Service für die Ver-
gabe 60
Mini Certificates 57
MQeAttribute 56
MQeMTrustAttribute 56
MQSeries, Kompatibilität mit 78
MQSeries, Schnittstelle zu 68
MQSeries-Brücke 10, 68
MQSeries-Brückenregeln 62
MQSeries-Client 9
MQSeries Everyplace,
Registrierungsdatenbank 17
MQSeries Everyplace-Adapter 49
MQSeries Everyplace-
Anwendungen 14
MQSeries Everyplace-Einheiten 17
MQSeries Everyplace-Gateways 17

MQSeries Everyplace-Klassen 65
MQSeries Everyplace-Netze 50, 67
MQSeries Everyplace-Objekte 18
MQSeries Everyplace-Regeln 61
MQSeries Everyplace-
Registrierungsdatenbank 57
MQSeries Everyplace-Sicherheit 52
MQSeries Everyplace-
Verwaltung 44
MQSeries Everyplace-
Warteschlangen 25
MQSeries Everyplace-WS-
Manager 32
MQSeries Integrator viii, 7
MQSeries-
Nachrichtenübertragung 7
MQSeries-Netze 67
MQSeries-Produktfamilie 7
MQSeries-Server 9
MQSeries Workflow viii, 7

N

Nachrichten, Verwaltung 44
Nachrichtenbasierte Sicherheit 55
Nachrichtenobjekte 18
Nachrichtenübertragung, asyn-
chron 50
Nachrichtenübertragung, MQSe-
ries 7
Nachrichtenübertragung, syn-
chron 51
Nachrichtenumsetzung 76
Nachrichtenzustellung, gesichert 80
Netze, MQSeries 67
Netze, MQSeries Everyplace 50, 67

O

Objekte, MQSeries Everyplace 18
Objekte, Nachrichten 18
Öffentliche Registrierungs-
datenbank 59

P

Peer-zu-Peer-Kanäle 10
Peer-zu-Peer-Verbindung 63
Private Registrierungsdatenbank 58
Produktkonzepte 17
Programmierschnittstellen 81
Proxy-WS-Managerobjekt 70

R

- Rechtliche Hinweise 91
- Regeln, MQSeries Everyplace 61
- Registrierungsdatenbank 57
- Registrierungsdatenbank, MQSeries Everyplace 17
- Registrierungsdatenbank, öffentlich 59
- Registrierungsdatenbank, privat 58
- Replikation von Zertifikaten 59

S

- Schnittstelle, Sicherheit 61
- Schnittstellen, Programmierung 81
- Schnittstellen zu MQSeries 68
- Server, MQSeries 9
- Sicherheit, auf Nachrichtenebene 55
- Sicherheit, auf Warteschlangenebene 54
- Sicherheit, lokal 53
- Sicherheit, MQSeries Everyplace 52
- Sicherheitsschnittstelle 61
- Softwareumgebungen 3
- Speichern und Weiterleiten, Warteschlangen 27
- Synchrone Nachrichtenübertragung 51
- Systemprogrammierschnittstelle 81

T

- Trace, MQSeries Everyplace 50

U

- Übersicht 1
- Übertragung 63
- Überwachung 47
- Umgebungen, Software 3
- Umsetzungsprogramme 76
- Universelle (Pervasive) Nachrichtenübertragung viii, 8
- Unterstützte Betriebssysteme 3

V

- Verbindung, Client/Server 64
- Verbindung, Peer-zu-Peer 63
- Verbindungsarten 63
- Verbindungsarten, mehrere 64
- Vergabeservice für Mini Certificates 60
- Verschlüsselung 52
- Verteilte Nachrichtenübertragung viii, 8
- Verwaltung mit MQSeries Everyplace 44
- Verwaltung von Einwählverbindungen 49

- Verwaltungsnachrichten 44
- Verwaltungswarteschlangen 29
- Voraussetzungen 3
- Vorgänge, WS-Manager 40
- Vorkenntnisse viii

W

- Warteschlangen, ferne 26
- Warteschlangen, lokale 25
- Warteschlangen, MQSeries Everyplace 25
- Warteschlangen, Speichern und Weiterleiten 27
- Warteschlangen, Verwaltung 29
- Warteschlangenbasierte Sicherheit 54
- Warteschlangenregeln 62
- WS-Manager 8, 10, 63
- WS-Manager, MQSeries Everyplace 32
- WS-Managerregeln 62
- WS-Managervorgänge 40

Z

- Zertifikatreplikation 59
- Zielgruppe vii
- Zu diesem Handbuch vii

Kommentare an IBM senden

Sie können uns Anmerkungen zu dem vorliegenden Handbuch über die nachfolgend aufgeführten Wege zukommen lassen.

Bitte lassen Sie es uns wissen, wenn Informationen Ihrer Meinung nach fehlerhaft sind oder ganz fehlen, oder wenn Sie Anmerkungen zur Richtigkeit, zum Aufbau, Inhalt oder zur Vollständigkeit des Handbuchs haben.

Bitte senden Sie uns Kommentare nur im Zusammenhang mit dem vorliegenden Handbuch und nur über die hier aufgeführten Übermittlungskanäle zu.

Wenn Sie Anmerkungen zu Funktionen von IBM Produkten bzw. IBM Systemen haben, wenden Sie sich bitte an Ihren IBM Ansprechpartner bzw. den zuständigen IBM Vertriebspartner.

Bei IBM eingehende Kommentare können von IBM beliebig verwendet werden, ohne dass hieraus eine Verpflichtung gegenüber dem Absender entsteht.

Ihre Kommentare können Sie IBM auf folgenden Wegen zukommen lassen:

- Per Post an folgende Adresse:
 - User Technologies Department (MP095)
 - IBM United Kingdom Laboratories
 - Hursley Park
 - WINCHESTER,
 - Hampshire
 - SO21 2JN
 - Großbritannien
- Per Fax:
 - Benutzer außerhalb von Großbritannien müssen im Anschluss an die jeweilige internationale Durchwahl (in Deutschland z. B. 00) folgende Nummer wählen: 44-1962-842327
 - Benutzer in Großbritannien müssen folgende Nummer wählen: 01962-842327
- Per E-Mail, unter Angabe der entsprechenden Netz-ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink: HURSLEY(IDRCF)
 - Internet: idrcf@hursley.ibm.com

Unabhängig von der Übertragungsart sind auf jeden Fall folgende Angaben erforderlich:

- Titel und Bestellnummer der Veröffentlichung
- Der Abschnitt, auf den Sie sich beziehen;
- Ihre Adresse: Name, Adresse, Telefonnummer, Faxnummer, Netz-ID.

Antwort

MQSeries Everyplace for Multiplatforms
Einführung
Version 1.2

IBM Form GC12-2860-02

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 01803/31 32 33) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: ibmterm@de.ibm.com

Name

Adresse

Firma oder Organisation

Rufnummer

E-Mail-Adresse

Antwort
GC12-2860-02



IBM Deutschland GmbH
SW TSC Germany

70548 Stuttgart



GC12-2860-02

