

MQSeries Everyplace for Multiplatforms



# Introduzione

*Versione 1.2*



MQSeries Everyplace for Multiplatforms



# Introduzione

*Versione 1.2*

## Attenzione

Prima di utilizzare queste informazioni e il relativo prodotto, leggere attentamente le informazioni generali in "Appendice. Informazioni particolari" a pagina 85

### Avvertenza sulle licenze

MQSeries Everyplace Versione 1.2 è un toolkit che consente agli utenti di scrivere applicazioni MQSeries Everyplace e di creare un ambiente in cui eseguirle.

Prima di installare questo prodotto o le relative applicazioni in un ambiente di produzione, accertarsi di disporre delle licenze necessarie.

Per utilizzare MQSeries Everyplace su piattaforme server specifiche (non per lo sviluppo di codici e per test), è necessario disporre delle *autorizzazioni all'utilizzo in base all'unità di capacità* (registrate nei documenti Proof of Entitlement e valide per supportare l'utilizzo di MQSeries Everyplace in base alle tabelle dei gruppi di politica prezzi e di unità di capacità pubblicate) in modo da poter utilizzare il programma su ciascuna macchina e suoi relativi aggiornamenti.

È necessario disporre delle *autorizzazioni all'utilizzo delle piattaforme dei dispositivi* (registrate nei documenti Proof of Entitlement e valide per supportare l'utilizzo di MQSeries Everyplace) per utilizzare il prodotto (non per lo sviluppo di codici e per test) su piattaforme client specifiche. Tali licenze non autorizzano l'utilizzo di MQSeries Everyplace Bridge oppure l'esecuzione sulle piattaforme server specificate negli elenchi dei prezzi di MQSeries Everyplace pubblicati dalla IBM e disponibili anche sul Web al seguente URL:

Per i dettagli relativi a queste limitazioni, visitare il sito all'indirizzo <http://www.ibm.com/software/mqseries> .

### Terza edizione (Maggio 2001)

Questa edizione si riferisce a MQSeries Everyplace Versione 1.2 e a tutti i rilasci e le modifiche successive, salvo diversamente indicato in nuove edizioni.

Questo documento viene aggiornato continuamente con nuove informazioni. Per l'ultima edizione, consultare la pagina Web della libreria di MQSeries all'indirizzo <http://www.ibm.com/software/mqseries/library/>.

© Copyright International Business Machines Corporation 2000, 2001. Tutti i diritti riservati.

# Indice

<b>Informazioni su questa pubblicazione . . . . .</b>	<b>v</b>
A chi si rivolge questa pubblicazione . . . . .	v
Prerequisiti . . . . .	v
Termini utilizzati in questa pubblicazione . . . . .	vi
<b>Sommario delle modifiche . . . . .</b>	<b>vii</b>
Modifiche apportate a questa edizione (GC13-2908-02) . . . . .	vii
Modifiche apportate all'edizione precedente (GC13-2908-01) . . . . .	vii
<b>Capitolo 1. Panoramica . . . . .</b>	<b>1</b>
<b>Capitolo 2. Ambienti software . . . . .</b>	<b>3</b>
Piattaforme supportate . . . . .	3
Ambiente Java . . . . .	4
Requisiti di memoria . . . . .	4
<b>Capitolo 3. Famiglia MQSeries . . . . .</b>	<b>7</b>
Prodotti distribuiti e per host di MQSeries . . . . .	8
MQSeries Everyplace . . . . .	9
<b>Capitolo 4. Requisiti del prodotto . . . . .</b>	<b>13</b>
Caratteristiche . . . . .	13
Applicazioni . . . . .	14
Requisiti del cliente . . . . .	14
<b>Capitolo 5. Contenuti del prodotto . . . . .</b>	<b>17</b>
Introduzione . . . . .	17
Oggetti messaggi . . . . .	18
Formato dei dati dump . . . . .	23
Code . . . . .	24
Programmi di gestione code. . . . .	31
Configurazione del programma di gestione code . . . . .	34
Operazioni dei programmi di gestione code . . . . .	38
Connessioni . . . . .	39
Amministrazione . . . . .	41
Messaggi di amministrazione . . . . .	41
Amministrazione selettiva . . . . .	44
Programma di controllo e operazioni relative. . . . .	44
Canali dinamici . . . . .	44
Adattatori . . . . .	46
Gestione dei collegamenti dialup . . . . .	46
Traccia . . . . .	46
Registrazione eventi . . . . .	47
Consegna dei messaggi . . . . .	47
Consegna asincrona dei messaggi . . . . .	47
Consegna sincrona dei messaggi . . . . .	48
Sicurezza . . . . .	48
Sicurezza locale MQSeries Everyplace . . . . .	50
Sicurezza MQSeries Everyplace basata sulle code . . . . .	50
Sicurezza a livello di messaggio . . . . .	52
Il registro . . . . .	53
Entità autenticabili MQSeries Everyplace . . . . .	54
Registro privato e credenziali . . . . .	54
Autoregistrazione . . . . .	54
Registro pubblico e replica di certificati . . . . .	55
Utilizzo applicazioni dei servizi di registro . . . . .	55
Servizi di creazione predefiniti di mini-certificati . . . . .	55
L'interfaccia di sicurezza . . . . .	56
Personalizzazione . . . . .	56
Regole . . . . .	56
Stili di collegamento . . . . .	58
Collegamento peer-to-peer . . . . .	59
Collegamento Client-server . . . . .	59
Stili di collegamento multipli . . . . .	59
Classi . . . . .	60
Caricamento applicazioni . . . . .	61
<b>Capitolo 6. Reti MQSeries Everyplace e MQSeries . . . . .</b>	<b>63</b>
Interfaccia per MQSeries . . . . .	63
Conversione messaggi . . . . .	71
Funzione . . . . .	73
Compatibilità . . . . .	73
Consegna assicurata . . . . .	74
<b>Capitolo 7. Interfacce di programmazione . . . . .</b>	<b>75</b>
<b>Capitolo 8. Introduzione a MQSeries Everyplace . . . . .</b>	<b>77</b>
Uso di MQSeries Everyplace . . . . .	78
Come acquisire esperienza . . . . .	79
Utilizzo di ES02: MQe_Explorer . . . . .	80

<b>Appendice. Informazioni particolari . . . . .</b>	<b>85</b>	<b>Bibliografia . . . . .</b>	<b>93</b>
Marchi . . . . .	87	<b>Indice analitico . . . . .</b>	<b>95</b>
<b>Glossario . . . . .</b>	<b>89</b>		

---

## Informazioni su questa pubblicazione

Questa pubblicazione costituisce un'introduzione generale a MQSeries Everyplace for Multiplatforms (definito MQSeries Everyplace). Tratta del contenuto del prodotto e delle relazioni con altri prodotti MQSeries.

Per informazioni dettagliate sull'API di MQSeries Everyplace e su come utilizzarla per creare applicazioni MQSeries Everyplace, consultare *MQSeries Everyplace for Multiplatforms Programming Reference* e *MQSeries Everyplace for Multiplatforms Programming Guide*.

Per informazioni relative all'utilizzo di altri linguaggi di programmazione con MQSeries Everyplace for Multiplatforms, consultare *MQSeries Everyplace for Multiplatforms Native Client Information*

Per le procedure di installazione di MQSeries Everyplace for Multiplatforms, consultare *MQSeries Everyplace for Multiplatforms - Readme*

Questo documento viene aggiornato continuamente con nuove informazioni. Per l'ultima edizione, consultare la pagina Web della libreria di MQSeries all'indirizzo <http://www.ibm.com/software/ts/mqseries/library/>.

---

## A chi si rivolge questa pubblicazione

Questa pubblicazione è rivolta a quegli utenti interessati a sistemi di messaggistica sicura su dispositivi leggeri come sensori, telefoni, PDA (Personal Digital Assistants) e computer laptop e a coloro che hanno bisogno di ampliare l'ambito della rete di messaggi di MQSeries Everyplace.

---

## Prerequisiti

Non è richiesta alcuna preparazione di base per leggere queste informazioni, ma una conoscenza iniziale dei concetti della messaggistica sicura costituirà un valido aiuto.

In caso contrario, si consiglia di leggere la seguente pubblicazione di MQSeries:

- *MQSeries An Introduction to Messaging and Queuing*

Questa pubblicazione è disponibile in linea presso la sezione delle pubblicazioni della libreria in linea di MQSeries. Si accede alla libreria dal sito Web di MQSeries all'indirizzo URL <http://www.ibm.com/software/ts/MQSeries/library/>

---

## Termini utilizzati in questa pubblicazione

I seguenti termini sono utilizzati nell'arco di questa pubblicazione:

### Famiglia MQSeries

si riferisce ai seguenti MQSeries prodotti:

- **MQSeries Workflow** semplifica l'integrazione di un'intera azienda con l'automazione dei processi aziendali, che coinvolga il personale e le applicazioni
- **MQSeries Integrator** è un software potente per la messaggistica che consente l'accodamento di messaggi in tempo reale e basato su criteri intelligenti, oltre che la trasformazione e la formattazione del contenuto
- **MQSeries Messaging** consente connessioni any-to-any da desktop a mainframe, attraverso un sistema di messaggistica aziendale con oltre 35 piattaforme supportate

### Messaggistica MQSeries

si riferisce ai seguenti gruppi di prodotti di messaggistica:

- **Messaggistica distribuita:** MQSeries per Windows NT, AIX, AS/400, HP-UX, Sun Solaris e altre piattaforme
- **Messaggistica host:** MQSeries per OS/390
- **Messaggistica diffusa:** MQSeries Everyplace

### MQSeries

si riferisce ai tre gruppi di prodotti di messaggistica MQSeries:

- Messaggistica distribuita
- Messaggistica host
- Messaggistica per workstation

### MQSeries Everyplace

Si riferisce al terzo MQSeries gruppo di prodotti di messaggistica, sulla messaggistica più diffusa.

### Piattaforma dispositivo

Un elaboratore di dimensioni ridotte, che consente di eseguire MQSeries Everyplace soltanto come client.

### Piattaforma server

Un elaboratore di qualsiasi dimensione che consente di eseguire MQSeries Everyplace come server oppure come client.

### Gateway

Un elaboratore di qualsiasi dimensione che consente di eseguire i programmi MQSeries Everyplace che includono la funzione MQSeries-bridge.

---

## Sommario delle modifiche

Questa sezione descrive le modifiche apportate a questa edizione di *Introduzione a MQSeries Everyplace for Multiplatforms*. Le modifiche sono contrassegnate da una riga verticale posta sulla sinistra.

---

### Modifiche apportate a questa edizione (GC13-2908-02)

---

### Modifiche apportate all'edizione precedente (GC13-2908-01)

sono state aggiunte le seguenti informazioni:

- Informazioni relative all'utilizzo di MQSeries Everyplace in ambiente AIX e Solaris.
- Requisiti di memoria.
- Modulo per i commenti del lettore.



---

## Capitolo 1. Panoramica

MQSeries Everyplace è un membro della famiglia di prodotti per la messaggistica aziendale MQSeries. È progettato per soddisfare le esigenze di messaggistica di dispositivi leggeri come sensori, telefoni, PDA (Personal Digital Assistants) e computer laptop, così come le apparecchiature mobili e le esigenze che riguardano l'impiego di reti di comunicazione di tipo non protetto. Il prodotto fornisce la qualità standard del servizio di MQSeries, fornendo la consegna assicurata per una volta sola, e consente di scambiare messaggi con altri prodotti della famiglia. Poiché sono molte le applicazioni MQSeries Everyplace che si eseguono al di fuori della protezione di un firewall su Internet, fornisce anche delle sofisticate funzioni di sicurezza.

I dispositivi leggeri richiedono che il sistema di messaggistica secondario utilizzi in modo accurato le risorse del sistema e, di conseguenza, MQSeries Everyplace offre funzioni su misura e interfacce adeguate all'impostazione dell'utente e non intende fornire esattamente le stesse funzioni degli altri membri della famiglia. D'altro canto, include una funzione univoca per supportare le classi particolari dell'utente, come, ad esempio, la sicurezza completa, gli oggetti messaggi, la messaggistica sincrona e asincrona, l'accesso remoto alla coda e i messaggi di Push e Pull.

Websphere Application Server è stato progettato per ben integrarsi con altri prodotti della famiglia di sistemi informatici IBM e altri componenti del Websphere Everyplace Server.



---

## Capitolo 2. Ambienti software

---

### **Piattaforme supportate**

MQSeries Everyplace è direttamente installabile solo su determinate piattaforme server. Per trasferire i programmi e le classi Java su altre piattaforme, è necessario utilizzare un programma di trasferimento file o di scaricamento (non fornito).

### **Piattaforme direttamente supportate con supporto di installazione**

Le seguenti piattaforme sono quelle su cui è possibile installare il prodotto utilizzando gli strumenti incorporati.

- Windows NT v4
- Windows 2000
- Windows 95/98/ME
- AIX Versione 4.3
- Sun Solaris Versione 7 o 8
- Linux Intel Kernel 2.2 (installato utilizzando un file zip).
- HP-UX 11.0 (installato utilizzando un file zip).

### **Piattaforme direttamente supportate prive del supporto di installazione**

Le seguenti piattaforme sono supportate per la verifica e la configurazione MQSeries Everyplace, ma supportano solo l'installazione tramite trasferimento file da un'altra piattaforma.

- WinCE 2.1 in esecuzione su dispositivi HP Jornada (Modelli 680 o 820)
- EPOC 32 bit Rilascio 5 in esecuzione su dispositivi Psion (5MX Pro o NetBook)
- PalmOS, V3.0 o successiva, in esecuzione su Palm V e IBM Workpad C3
- IBM 4690 OS con Java

### **Piattaforme supportate indirettamente**

È possibile utilizzare le seguenti piattaforme, ma sono supportate solo se il relativo ambiente Java è completamente compatibile con quello sulle piattaforme direttamente supportate. I problemi possono essere esaminati solo su una delle piattaforme elencate sopra.

- Linux su zSeries in esecuzione su Kernel 2.2
- iSeries
- OS/2

## ambienti software

- EPOC (su dispositivi diversi da quelli elencati sopra)
- WinCE (su dispositivi diversi da quelli elencati sopra)
- QNX Neutrino
- Pocket PC
- PalmOS (su dispositivi diversi da quelli elencati sopra)
- Qualsiasi altra piattaforma che esegue uno degli ambienti Java elencati in "Ambiente Java"

---

### Ambiente Java

È richiesto uno dei seguenti ambienti runtime Java:

- IBM Java runtime (JVM 1.3 o successivo), incluso Java Micro Edition
- In qualsiasi ambiente Java che sia Sun Java (V1.1 o successivo) certificato <sup>1</sup>

**Nota:** È necessario che Java sia completamente compatibile con gli ambienti testati su una delle seguenti piattaforme in modo che il servizio sia disponibile

- I dispositivi HP Jornada (Modelli 680 o 820) in esecuzione sul sistema operativo Windows CE
- Dispositivi Psion (5MX Pro o NetBook) in esecuzione sul sistema operativo EPOC
- Una delle piattaforme server comprese nell'elenco delle piattaforme supportate direttamente

Per il funzionamento di MQSeries-bridge, è necessario disporre di MQSeries Classes per Java. Controllare il livello Java necessario per eseguire la versione MQSeries Classes per Java.

---

### Requisiti di memoria

La seguente tabella indica la memoria necessaria per eseguire l'installazione di MQSeries Everyplace.

*Tabella 1. Memoria richiesta per eseguire l'installazione*

Sistema operativo	Memoria richiesta
Windows NT (file system = NTFS)	26Mb
AIX	29Mb
Solaris	27Mb

---

1. È possibile che si verifichino problemi se si esegue il programma di installazione Sun JVM con il compilatore JIT (Just In Time) abilitato. Se si utilizza Sun JVM, disabilitare il compilatore JIT utilizzando il comando: `java -Djava.compiler=NONE install`

La seguente tabella indica la memoria necessaria per i file MQSeries Everyplace dopo aver eseguito l'installazione.

*Tabella 2. Memoria richiesta per MQSeries Everyplace*

<b>Sistema operativo</b>	<b>Memoria richiesta</b>
Windows NT (file system = NTFS)	9,5Mb
AIX	11Mb
Solaris	10Mb



---

## Capitolo 3. Famiglia MQSeries

La famiglia MQSeries comprende una serie di prodotti che offrono un'ampia gamma di funzioni, come illustrato in Figura 1

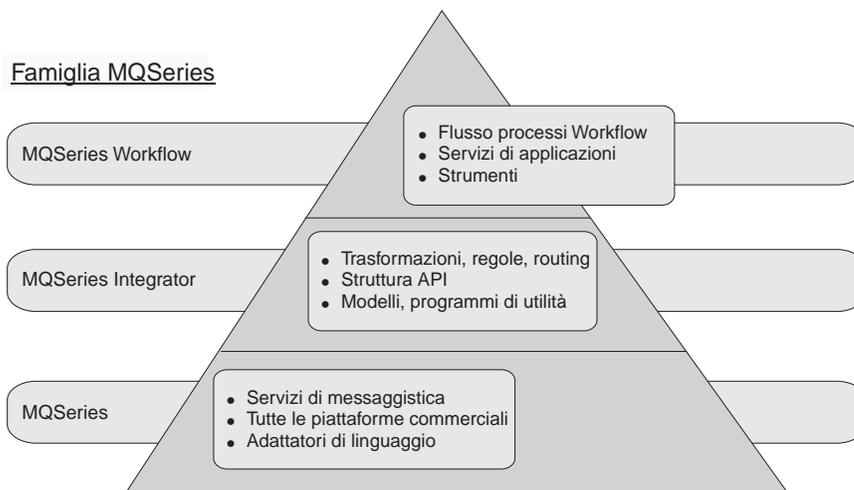


Figura 1. Famiglia MQSeries

- **MQSeries Workflow** semplifica l'integrazione di un'intera azienda con l'automazione dei processi aziendali, che coinvolga il personale e le applicazioni
- **MQSeries Integrator** è un software potente per la messaggistica che consente l'accodamento di messaggi in tempo reale e basato su criteri intelligenti, oltre che la trasformazione e la formattazione del contenuto
- **MQSeries Messaging** consente connessioni any-to-any da desktop a mainframe, attraverso un sistema di messaggistica aziendale con oltre 35 piattaforme supportate

Sia MQSeries Workflow che MQSeries Integrator sono prodotti che sfruttano le connessioni fornite dal sistema di messaggistica di MQSeries.

La messaggistica di MQSeries è fornita dai prodotti MQSeries e MQSeries Everyplace ; ognuno è progettato per supportare una o più piattaforme hardware server e/o sistemi operativi associati. Data la gran varietà di funzioni di piattaforma, questi singoli prodotti sono organizzati in gruppi e riflettono funzioni e progetti comuni. Esistono tre gruppi di prodotti:

## Famiglia MQSeries

- **Messaggistica distribuita:** MQSeries per Windows NT, AIX, AS/400, HP-UX, Sun Solaris e altre piattaforme
- **Messaggistica host:** MQSeries per OS/390
- **Messaggistica diffusa:** MQSeries Everyplace

La messaggistica si basa su gestori code, a prescindere dal singolo prodotto o gruppo di prodotti. Questi programmi gestiscono code che a loro volta possono contenere messaggi. Le applicazioni comunicano con un programma di gestione code locale e ricevono o memorizzano i messaggi sulle code. Se un messaggio viene memorizzato su una coda remota, gestita da un relativo programma remoto, il messaggio verrà trasmesso su canali al gestore code remoto. In questo modo i messaggi possono passare attraverso vari programmi di gestione intermedi prima di arrivare a destinazione. L'obiettivo della messaggistica consiste nel separare l'applicazione di invio a quella di ricezione, accodando i messaggi a punti intermedi, se necessario. Tutti i prodotti di messaggistica di MQSeries riguardano gli stessi elementi di base per gestori di code, code, messaggi e canali, nonostante le tante differenze nel dettaglio.

---

### Prodotti distribuiti e per host di MQSeries

I prodotti MQSeries per host e messaggistica distribuita sono utilizzati per supportare varie configurazioni di rete diverse, tutte riguardanti client e server.<sup>2</sup> Alcuni esempi sono illustrati in Figura 2.

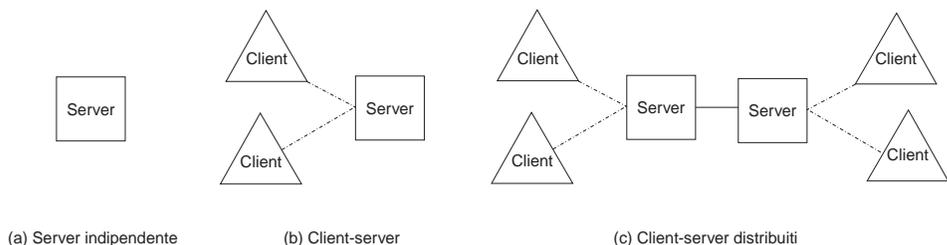


Figura 2. Configurazioni semplici per host e distribuite

Nel caso più semplice un server indipendente viene configurato eseguendo un programma di gestione code. Sul server vengono eseguite una o più applicazioni, scambiando messaggi tramite code. Una configurazione alternativa è quella client-server, in cui il programma di gestione code esiste solo sul server, ma ciascun client ne ha accesso tramite un *canale client*. Il canale client è un collegamento per comunicazioni bidirezionali che funziona con un protocollo MQSeries unico, implementando qualcosa di simile a un

---

2. Questi termini hanno un significato specifico all'interno di MQSeries che non sempre corrisponde al loro utilizzo più diffuso.

RPC (Remote Procedure Call). È possibile eseguire le applicazioni sui client e accedere alle code dei server. Un vantaggio della configurazione client-server è dato dal fatto che l'infrastruttura della messaggistica client è leggera, in quanto dipende dal programma di gestione code del server. Uno degli svantaggi, invece, consiste nel fatto che i client e il relativo server funzionano sincronicamente, per cui il canale del client deve essere sempre disponibile.

La configurazione distribuita client-server mostra un caso più complesso, riguardante più server. In queste configurazioni i server scambiano messaggi attraverso appositi *canali per i messaggi*. I canali per i messaggi sono unidirezionali, con un protocollo studiato per uno scambio sicuro e asincrono dei dati di messaggi. Non è necessario che questi canali per i messaggi siano disponibili ai client per continuare a funzionare, ma nessun messaggio può essere trasmesso tra i server quando le comunicazioni non sono disponibili.

---

### MQSeries Everyplace

MQSeries Everyplace supporta un'ampia serie di configurazioni di rete attraverso l'impiego dei programmi di gestione code, con ciascuno di essi abilitato ad appropriate funzioni. Non esiste alcun concetto di client o server come nei prodotti distribuiti e per host di MQSeries. I gestori di code di MQSeries Everyplace possono comportarsi come client o server di tipo tradizionale, ma, in realtà, ognuno di essi è semplicemente un programma di gestione code abilitato ad eseguire determinate attività di applicazione. A dimostrazione di ciò, è possibile configurare un programma di gestione di MQSeries Everyplace con o senza code locali. Con le code locali è possibile memorizzare i messaggi in locale e offrire la messaggistica asincrona delle applicazioni, mentre senza le code locali tutto ciò viene limitato alla messaggistica sincrona. Un altro esempio di configurazione adatta riguarda la possibilità di configurare un gestore code con o senza le funzioni di bridge. Con il bridge è possibile scambiare messaggi con i gestori di code della workstation e dell'host di MQSeries; senza il bridge, invece, è possibile comunicare direttamente soltanto con altri programmi di gestione code di MQSeries Everyplace (sebbene sia possibile comunicare indirettamente attraverso altri programmi di gestione code nella rete che dispone delle funzioni del bridge).

**Nota:** un nuovo nodo per MQSI (MQSeries Integrator) consente la connessione di MQSeries Everyplace direttamente, senza l'utilizzo di MQSeries-bridge.

I programmi di gestione code di MQSeries Everyplace utilizzano *canali dinamici* per scambiare informazioni. Tali canali dinamici dispongono di diverse proprietà dai *canali client* e dai *canali per i messaggi* utilizzati da altri membri della famiglia. Di seguito vengono riportate alcune delle funzioni principali dei canali dinamici:

## Famiglia MQSeries

- Supporto sia per la messaggistica *sincrona* che *asincrona*: la messaggistica sincrona fornisce un servizio di trasmissione direttamente dall'applicazione di origine alla coda di destinazione, senza l'accodamento nel programma di gestione code di origine. La messaggistica asincrona è un servizio di trasmissione dal programma di gestione code di origine alla coda di destinazione, con possibile accodamento nel programma di gestione code di origine.
- Impiego del *servizio End-to-end*: i canali vanno dal programma di gestione code di origine ad uno di destinazione, passando possibilmente per i programmi di gestione intermedi. Il protocollo di trasporto utilizzato potrebbe modificarsi quando il canale passa attraverso i programmi intermedi.
- Supporto per la *compressione*, la *codifica* e l'*autenticazione*: i canali dispongono di tali caratteristiche di sicurezza per proteggere i dati in transito.
- Supporto per le operazioni *peer-to-peer* e *client-server*: i canali definiti peer-to-peer sono funzionalmente simmetrici, in modo tale che o l'origine o la destinazione possano inizializzare un'operazione sul canale. I canali Client-server sono di tipo richiesta/risposta: il client effettua una richiesta al server e il server risponde ad essa. (Si noti che ciò non limita il flusso del messaggio. È possibile inviare messaggi dal client al server e viceversa).

L'opzione relativa alla configurazione del bridge consente ad un programma di gestione code di MQSeries Everyplace di comunicare con i programmi di gestione code distribuiti e per host di MQSeries attraverso i *canali client*. Il componente del bridge gestisce un gruppo di canali client che è possibile collegare a uno o più programmi di gestione code distribuiti o per host. È possibile configurare i programmi di gestione code di MQSeries Everyplace abilitati al bridge multiplo in una sola rete per fornire la capacità, l'esecuzione e l'affidabilità richieste.

Alcune configurazioni tipiche, ma arbitrarie, di MQSeries Everyplace vengono illustrate nei seguenti diagrammi. Per maggior chiarezza i diagrammi illustrano soltanto le connessioni dirette definite. Possono essere definite, inoltre, anche le connessioni indirette che sfruttano le connessioni dirette. Per rappresentare un canale peer-to-peer viene utilizzata una doppia freccia, mentre per un canale client-server una sola freccia rivolta verso il server. I Client possono utilizzare il canale client-server sia per inviare messaggi al server che per ricevere messaggi da quello stesso server. Le linee senza freccia indicano i canali client di MQSeries che abilitano le comunicazioni tra MQSeries Everyplace e MQSeries.



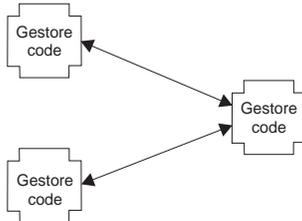
(a) Gestore code indipendente

(a) Illustra un programma di gestione code indipendente utilizzato per supportare una o più applicazioni che utilizzano le code per lo scambio di dati.



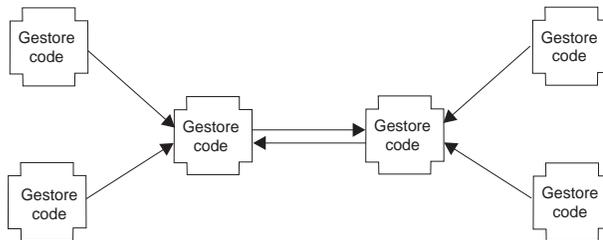
(b) Configurazione peer-to-peer

(b) Illustra due programmi di gestione code interconnessi su un canale peer-to-peer.



(b) Configurazione peer-to-peer

(c) Illustra un secondo canale peer-to-peer diretto ad un terzo programma di gestione code. Tutti e tre possono sfruttare le connessioni dirette sottostanti per scambiarsi i dati.

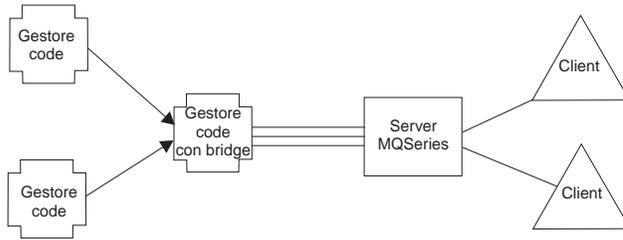


(d) Una piccola rete

(d) Illustra una piccola configurazione di rete, in cui i programmi di gestione code del server centrale utilizzano un paio di canali client-server diretti per scambiarsi informazioni. Ciascun programma di gestione code del client utilizza un canale client-server diretto con uno dei programmi di gestione

## Famiglia MQSeries

code del server.



(e) Una rete integrata della famiglia MQSeries

(e) Illustra una configurazione di MQSeries Everyplace in cui uno dei gestori code è stato configurato con l'opzione del bridge e il gruppo di canali client è stato diretto verso un singolo server di destinazione distribuito e per host di MQSeries.

---

## Capitolo 4. Requisiti del prodotto

Questo capitolo descrive i requisiti che caratterizzano la progettazione e l'implementazione di MQSeries Everyplace.

---

### Caratteristiche

MQSeries Everyplace estende l'ambito della messaggistica alla famiglia MQSeries:

- Con il supporto di dispositivi low-end, come PDA, telefoni e sensori. In questo modo è possibile partecipare a una rete di messaggistica MQSeries. MQSeries Everyplace, inoltre, supporta i dispositivi di media dimensione, come laptop, workstation e piattaforme distribuite e per host. MQSeries Everyplace offre la stessa qualità di servizi, con unica consegna sicura dei messaggi, e consente lo scambio di messaggi con altri membri della famiglia.
- Offrendo servizi di messaggistica leggera.
- Fornendo ampie funzioni di sicurezza per proteggere messaggi, code e dati correlati, in memoria o in trasmissione.
- Funzionando in modo efficiente in ambienti di comunicazione ostili in cui le reti sono instabili, o in cui la larghezza di banda è limitata. MQSeries Everyplace dispone di un protocollo di rete efficiente e della correzione automatica di errori di comunicazione.
- Con il supporto dell'utente mobile, per fare in modo che i punti di connessione della rete vengano modificati in roaming dei dispositivi. MQSeries Everyplace consente, inoltre, di controllare il funzionamento in condizioni in cui le risorse della batteria e le reti sono limitate o riportano errori.
- Operando tramite firewall correttamente configurati
- Con la riduzione delle attività di gestione per l'utente, per poter tenere nascosta la presenza di MQSeries Everyplace su un dispositivo. Ciò rende MQSeries Everyplace una base ideale per creare applicazioni per programmi di utilità.
- Con una facile procedura di personalizzazione e di ampliamento, attraverso l'impiego di regole fornite dalle applicazioni e altre classi che modificano il funzionamento, oppure tramite una suddivisione delle classi di oggetti base, ad esempio, per rappresentare vari tipi di messaggi.

## requisiti del prodotto

---

### Applicazioni

La scelta delle applicazioni disponibili di MQSeries Everyplace è ampia e varia, ma una gran parte delle applicazioni può essere personalizzata per gruppi particolari di utenti. L'elenco riportato di seguito contiene alcuni esempi delle applicazioni considerate:

- **Applicazioni per la vendita al dettaglio:** compensazione delle transazioni di cassa ai sistemi host, come gli intermediari di messaggi
- **Applicazioni per il consumatore:** spesa al supermercato da casa utilizzando una PDA, raccolta delle opinioni dei viaggiatori sulle compagnie di voli, transazioni finanziarie da un telefono mobile
- **Applicazioni per il controllo:** raccolta e integrazione di dati da sensori di tubi dell'olio trasmessi via satellite, operazioni remote di attrezzature (per esempio valvole) con sicurezza per garantire la validità dell'operatore
- **Lavoratori in movimento:** agenti rappresentanti (ramo assicurazioni), stampa veloce di ricevute al cliente per società di recapiti, camerieri di fast-food in comunicazione con la cucina, punteggio ai tornei di golf, sistemi mobili protetti, sistemi di messaggistica per la polizia, informazioni per i lavoratori con difficoltà di comunicazione, lettura metrica di superfici.
- **Produttività personale:** replica posta/calendario, replica database, riduzione dimensioni laptop

---

### Requisiti del cliente

I requisiti che hanno influenzato la progettazione di MQSeries Everyplace comprendono:

- **Gestione:** gestione e impostazione minima; supporto per la gestione locale e remota; capacità di ampliare e personalizzare le funzioni amministrative per soddisfare le esigenze di applicazioni particolari; enfasi sulla scoperta e correzione automatica di problemi; elementi di amministrazione indipendenti da poter utilizzare in modo selettivo.
- **Comunicazioni:** un protocollo di rete molto efficiente; intestazioni minime; nessun campo obbligatorio nei messaggi (ad eccezione di un solo identificativo); capacità di modificare la codifica dei dati; supporto per compressione, codifica e autenticazione; negoziazione end-to-end delle funzioni di compressione e sicurezza; capacità di attraversare facilmente i firewall; adattatori di comunicazioni collegabili.
- **Compatibilità:** qualità MQSeries di servizi e interscambio di messaggi; capacità di comunicare con sistemi MQSeries esistenti senza modificare applicazioni; controllo flessibile dell'interscambio di messaggi tra MQSeries e MQSeries Everyplace.
- **Struttura:** per il dispositivo Palm, 64K byte. Per i dispositivi Java una dimensione minima dei file di classe di 100K byte.

- **Funzioni:** funzioni di messaggistica sincrona e asincrona, accesso ai messaggi memorizzati su code locali o remote; capacità di utilizzare tutti i campi di un messaggio per il recupero selettivo; controllo della funzione di memoria di una coda.
- **Supporto delle regole:** controllo di vari aspetti del funzionamento generale attraverso regole, ad esempio quando inviare messaggi, quando ritentare una comunicazione, cosa fare di un messaggio troppo grande, cosa fare quando una coda di destinazione è piena.
- **Sicurezza:** supporto completo per sicurezza, autenticazione e accettazione; sicurezza per livelli di code e di messaggi; protezione del sistema di messaggistica da interferenze alla sicurezza; sicurezza di collegamento tramite algoritmi standard del settore; capacità di integrazione con le credenziali del sistema operativo dell'utente: capacità di aderire ai requisiti di sicurezza nazionale, in modo che il supporto per la sicurezza venga modificato quando i messaggi passano i confini del paese.

## requisiti del prodotto

## Capitolo 5. Contenuti del prodotto

### Introduzione

Gli elementi fondamentali del modello di programmazione di MQSeries Everyplace sono messaggi, code e programmi di gestione code. I messaggi di MQSeries Everyplace sono oggetti contenenti dati definiti dalle applicazioni. Vengono memorizzati in una coda, per poi viaggiare su una rete MQSeries Everyplace. I messaggi sono indirizzati ad una coda di destinazione, specificando il relativo programma di gestione code e la coppia di nomi. Le applicazioni dispongono i messaggi sulle code con un'operazione di inserimento (put) e in genere li recuperano con un'operazione di ricezione (get). Le code possono essere locali o remote e sono gestiti da appositi programmi di gestione code. I dati della configurazione sono memorizzati in un registro.

La struttura dell'oggetto di MQSeries Everyplace può essere visualizzata nel seguente modo dallo strumento di gestione di MQE\_Explorer:

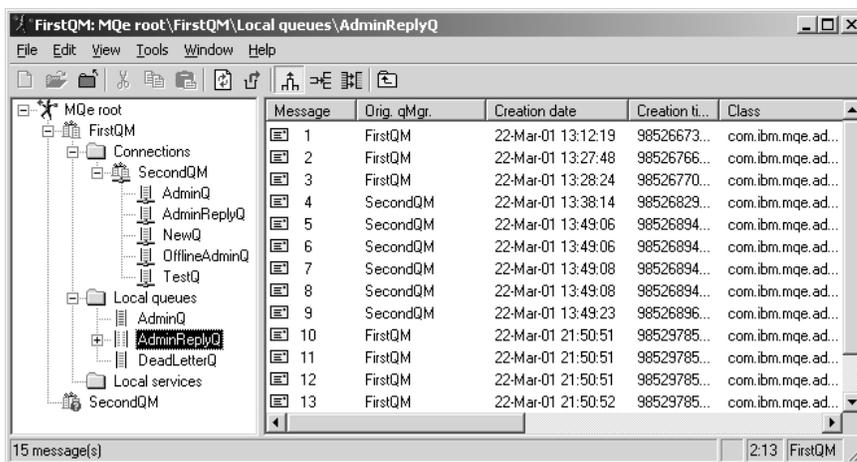


Figura 3. Gli oggetti di MQSeries Everyplace visualizzati da MQE\_Explorer

Nel riquadro a sinistra gli oggetti vengono visualizzati in una struttura ad albero. Al di sotto della radice di MQe sono presenti due programmi di gestione code, ExampleQM e OfficeQM. La struttura di OfficeQM viene espansa per mostrare le cartelle per le *connessioni*, le *code locali* e i *servizi locali*.

L'espansione delle *connessioni* mostra una definizione della connessione dal programma di gestione code OfficeQM a quello ExampleQM. Il programma di gestione code remoto, OfficeQM dispone di definizioni per due delle code di

## contenuti del prodotto

ExampleQM - AdminQ e AdminReplyQ. L'espansione delle *code locali* per OfficeQM mostra cinque code locali, una delle quali viene selezionata e visualizza i relativi contenuti nel riquadro a destra.

La comprensione di MQSeries Everyplace richiede una valutazione della natura dei messaggi, delle code, dei programmi di gestione code e delle connessioni. Questi concetti vengono delineati nelle seguenti sezioni.

---

### Oggetti messaggi

Gli oggetti messaggi di MQSeries Everyplace differiscono molto dai messaggi supportati dalla messaggistica di MQSeries. In MQSeries i messaggi sono dei gruppi di byte, divisi in un'intestazione e un corpo del messaggio. L'intestazione viene individuata e comprende informazioni fondamentali, come l'identità della risposta alla coda, la risposta al programma di gestione code, l'ID di messaggio e l'ID di correlazione. Il corpo del messaggio non viene individuato.

Al contrario, i messaggi in MQSeries Everyplace sono *oggetti messaggi*, ereditati da un oggetto di MQSeries Everyplace definito *oggetto campi*. Questi messaggi non dispongono di un'intestazione e di un corpo, ma di proprietà e metodi. La comprensione dell'oggetto messaggio richiede prima una valutazione del precedente oggetto campi.

Questi oggetti campi, utilizzati ampiamente in MQSeries Everyplace, sono un insieme di *campi*, in cui un campo è costituito da un nome, da un tipo di dati e dai dati stessi. I nomi dei campi sono stringhe in caratteri Ascii (un numero di caratteri protetti) di lunghezza illimitata.

Gli oggetti campi sono dotati di un *tipo*, in cui il tipo è una stringa abbreviata che corrisponde al nome classe dell'oggetto di programmazione. I campi possono essere di tipo:

Tabella 3. Tipi di oggetto campi

Tipo	Descrizione
Ascii	Stringa o un gruppo dinamico di stringhe Ascii
Valore Booleano	Valore
Valori Byte	Gruppo fisso o un gruppo dinamico di valori di byte
Valore a doppia virgola mobile	Valore, gruppo fisso o un gruppo dinamico di valori a doppia virgola mobile

Tabella 3. Tipi di oggetto campi (Continua)

Tipo	Descrizione
<b>Campi</b>	Oggetto campi o un gruppo dinamico di oggetti campi (nidificazione di oggetti campi supportata)
<b>Valore a virgola mobile</b>	Valore, gruppo fisso o un gruppo dinamico di valori a virgola mobile
<b>Cifra</b>	Valore (4 byte), gruppo fisso o un gruppo dinamico di cifre
<b>Cifra lunga</b>	Valore (8 byte), gruppo fisso o un gruppo dinamico di cifre lunghe
<b>Oggetto</b>	Valore
<b>Cifra breve</b>	Valore (2 byte), gruppo fisso o un gruppo dinamico di cifre brevi
<b>UNICODE</b>	Stringa o un gruppo dinamico di stringhe UNICODE

Gli oggetti campi supportano varie chiamate di metodo, come ad esempio:

- Enumerazione dei campi
- Copia di uno o più campi costitutivi (con l'opzione di sostituzione)
- Comparazione degli oggetti campi
- Inserimento e ricezione nei e dai campi costitutivi
- Esaminare gli oggetti campi (per esempio, 'l'oggetto contiene un campo costitutivo?')
- Eseguire il dump e il restore in e da un gruppo di byte

I metodi dump e restore sono fondamentali. Ad esempio, il metodo dump viene utilizzato per fornire i dati per la trasmissione di un oggetto messaggio in un collegamento e il metodo restore consente di ricreare quell'oggetto nel ricevente. In questo modo ciascun oggetto è responsabile del proprio formato di trasmissione e i metodi predefiniti forniti possono essere sovrascritti per implementare un funzionamento personalizzato. Allo stesso modo, questi due metodi vengono utilizzati quando gli oggetti messaggi vengono salvati o ripristinati dalle code. Con la sovrascrittura di tali metodi è possibile implementare differenti funzionamenti. Ad esempio, un oggetto messaggio può interrogare un database al momento del dump per estrarre il valore precedente alla trasmissione.

Nella seguente tabella sono indicate alcune delle proprietà degli oggetti campi e dei relativi campi costitutivi:

## oggetti messaggi

Tabella 4. Oggetti campi e proprietà

Proprietà	Presenza	
	Oggetti campi	Campi
Oggetto attributo associato	Facoltativo	
Campi costitutivi	sì	
Nascosto		sì
Nome		sì
Tipo	sì	sì
Valore		sì

Le due proprietà principali sono la proprietà *Hidden*, che consente di ignorare un campo per scopi di comparazione, e la possibilità di *associare un oggetto attributo* ad un oggetto campi.

Gli *oggetti attributo* sono fondamentali per il modello di sicurezza di MQSeries Everyplace e consentono una serie di funzioni, tra cui l'accesso selettivo al contenuto e la protezione di quest'ultimo. Dispongono delle seguenti importanti proprietà:

Tabella 5. Proprietà degli oggetti attributi

Proprietà	Descrizione
<b>Autenticazione:</b>	Accesso dei controlli
<b>Codifica:</b>	Protegge i dati quando si esegue il dump (e ne consente il recupero)
<b>Compressione:</b>	Riduce i requisiti della memoria (per operazioni di trasmissione e/o memorizzazione)
<b>Regola:</b>	Controlla le operazioni consentite

Gli oggetti messaggi ereditano dagli oggetti campi tutte le proprietà sopra indicate, inclusa la possibilità di disporre di oggetti attributi associati al messaggio e al livello dell'oggetto campi del componente. Inoltre, gli oggetti messaggi comprendono un UID (Unique Identifier), creato da MQSeries Everyplace. Tale UID identifica unicamente ogni oggetto messaggio nell'intera rete di MQSeries Everyplace ed è costituito da:

- **Nome** del programma di gestione code di origine (aggiunto dal programma di gestione code alla ricezione dell'oggetto). Deve trattarsi di un nome unico.
- **Ora** della creazione dell'oggetto messaggio (aggiunto alla creazione)

Pertanto, un oggetto messaggio dispone delle seguenti proprietà, oltre a quelle ereditate dal precedente oggetto campi:

*Tabella 6. Proprietà degli oggetti messaggi*

Proprietà	Descrizione
<b>Programma di gestione code di origine</b>	Il nome del programma di gestione code che per primo ha ricevuto il nuovo messaggio dall'applicazione
<b>Tempo per la creazione</b>	L'ora in cui l'applicazione ha creato l'oggetto messaggio
<b>UID</b>	L'unico identificatore di MQSeries Everyplace

Non è richiesta alcuna informazione in un messaggio destinato a un altro programma di gestione code di MQSeries Everyplace, anche se in genere sono presenti altri campi. Questi campi supplementari si presentano in vari modi:

- Campi aggiunti da MQSeries Everyplace per indicare lo stato corrente
- Campi associati ad una particolare classe secondaria di messaggi
- Campi personalizzati associati ad un esempio di oggetto messaggio

MQSeries Everyplace aggiunge i campi ad un oggetto messaggio (e in seguito li rimuove) per implementare le operazioni di messaggistica e di accodamento. Ad esempio, se un messaggio è su una coda, è possibile interrogarlo per scoprire quando è stato inserito su quella coda. (È possibile effettuare tale operazione utilizzando una regola della coda, successivamente illustrata). Quando un messaggio viene inviato tra i programmi di gestione code, può essere aggiunto un campo rinviato per indicare che è in corso una ritrasmissione dei dati. Esistono molti altri esempi.

Questi messaggi basati sulle applicazioni sono esempi che derivano dalla classe degli oggetti messaggi di base e, pertanto, contengono dei campi aggiuntivi (ad esempio, il numero della fattura). Alcuni di questi campi supplementari sono generici e comuni a molte applicazioni, come, ad esempio, il nome del programma di gestione code di risposta. MQSeries Everyplace fornisce il supporto per i campi che potrebbero essere presenti nei messaggi. Questi campi comprendono:

*Tabella 7. Campi di oggetti messaggi forniti*

Nome campo	Utilizzo
<b>Azione</b>	Utilizzato dall'amministrazione per indicare azioni come l'indagine, la creazione e la cancellazione
<b>ID di correlazione</b>	Stringa di byte utilizzata per correlare una risposta al messaggio originale

## oggetti messaggi

Tabella 7. Campi di oggetti messaggi forniti (Continua)

Nome campo	Utilizzo
<b>Errori</b>	Utilizzato dall'amministrazione per riportare le informazioni relative agli errori
<b>Orario di scadenza</b>	Ora in cui il messaggio può essere cancellato (anche se non recapitato)
<b>ID blocco</b>	La chiave necessaria per sbloccare un messaggio
<b>ID di messaggio</b>	Un unico identificatore per un messaggio
<b>Programma di gestione code di origine</b>	Il nome del programma di gestione code che ha inviato il messaggio
<b>Parametri</b>	Utilizzati dall'amministrazione per fornire i relativi dettagli
<b>Priorità</b>	Priorità relativa per la trasmissione dei messaggi
<b>Motivo</b>	Utilizzato dall'amministrazione per riportare le informazioni relative agli errori
<b>Coda di risposta</b>	Nome della coda a cui inviare la risposta al messaggio
<b>Gestore code di risposta</b>	Nome del programma di gestione code a cui inviare la risposta a un messaggio
<b>Rinvio</b>	Indica che si tratta del rinvio di un messaggio precedente
<b>Codice di errore</b>	Utilizzato dall'amministrazione per riportare lo stato di un'operazione di amministrazione
<b>Stile</b>	Distingue i comandi da richieste, risposte e via di seguito
<b>Impacchettamento messaggio</b>	Messaggio impacchettato per assicurare la protezione dei dati

In tutti i casi è disponibile una costante che consente di portare il nome campo a un solo byte. Per alcuni campi si adotta un provvedimento più esteso. Per esempio, la priorità (se presente) influisce sull'ordine in cui vengono trasmessi i messaggi; l'ID di correlazione provoca l'indicizzazione di una coda per consentire un recupero veloce dei valori dei campi; il tempo di scadenza avvia la scadenza del messaggio e così via.

Il pannello amministrativo illustrato in Figura 4 a pagina 23 indica la struttura di un messaggio. Si noti che l'utilizzo dei campi nidificati implica che all'interno dei messaggi possano sussistere complicate gerarchie dei dati e che è possibile approfondire gli strumenti amministrativi (e le applicazioni) nella struttura autodescrittiva, purché questa non sia stata bloccata tramite le impostazioni di sicurezza.

Gli oggetti messaggi utilizzano i metodi dump e restore quando sono serializzati in o da un gruppo di byte per la trasmissione o la memoria delle code. Per impostazione predefinita, allo scopo di salvarne la struttura, non trasmette la definizione della classe associata. Sovrascrivendo il dump e il restore con la serializzazione standard i metodi trasmettono la definizione della classe, qualora venga richiesto.

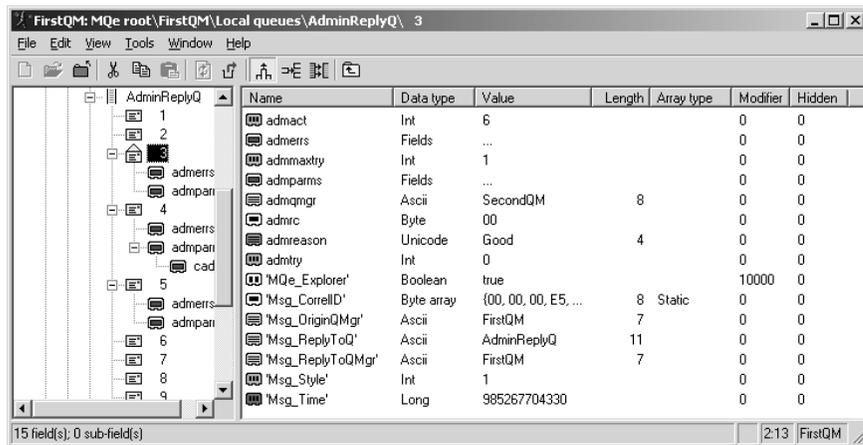


Figura 4. La struttura del messaggio visualizzato con MQE\_Explorer

Tuttavia, la classe viene resa disponibile indipendentemente per tutti i gestori di code in cui deve essere creata un'istanza dell'oggetto messaggio. Se è necessario creare un'istanza di un oggetto (per esempio per la memoria intermedia durante la trasmissione) ed è noto che il file di classe non è disponibile, il messaggio può essere impacchettato in un'altra classe. Questa è la tipica classe predefinita dell'oggetto messaggio. Tale tecnica, inoltre, risulta utile laddove un oggetto attributi è stato utilizzato per proteggere il messaggio. La presenza di un tale oggetto attributi implica che i contenuti del messaggio non sono accessibili senza le necessarie chiavi di sicurezza. Tuttavia impacchettare questo tipo di messaggio consente al messaggio kernel di essere protetto e nel contempo si può tranquillamente effettuare il dump e il ripristino del messaggio impacchettato.

Il metodo dump dell'oggetto messaggio predefinito è stato ottimizzato per ridurre la dimensione della stringa di byte generata ed avere così una memoria messaggi e una trasmissione efficienti.

## Formato dei dati dump

Il formato predefinito dei dati dump codifica i campi nel modo seguente:

```
{Lunghezza Identificativo Margine {Dati}} {Lunghezza
Identificativo Margine {Dati}} { ...}
```

## oggetti messaggi

dove:

- *Dati*: il valore dei dati. Le cifre sono compresse, con 0 e F rimossi. Gli indicatori booleani non hanno byte di dati associati
- *Margine*: un byte speciale che delimita il confine tra l'identificativo e la voce facoltativa Dati. Il byte indica anche il tipo di voce Dati
- *Identificativo*: contiene il nome campo di una stringa di byte Ascii di lunghezza variabile, che termina con un byte finale
- *Lunghezza*: indica la lunghezza del campo di dati. Si utilizza un numero variabile di byte tra 1 e 4. Il primo byte riserva i primi due bit per indicare la lunghezza del campo per la lunghezza. Sono supportate le lunghezze nell'intervallo tra 0 e 1.073.741.823

Ne deriva un flusso di dati altamente ridotto. È possibile eseguire altri salvataggi comprimendo i dati. La compressione XOR con un flusso di byte precedente può creare buoni risultati ma, data la natura variabile di tali campi e dal momento che l'ordine dei campi può variare, una semplice compressione XOR non sempre può dare l'effetto desiderato. MQSeries Everyplace comprende un XOR intelligente, che funziona campo per campo e che può migliorare la compressione.

---

## Code

Le code vengono utilizzate in genere per contenere gli oggetti messaggi in attesa di rimozione dai programmi di applicazioni. Come i messaggi, anche le code derivano dagli oggetti campi. Di norma, non è consentito l'accesso diretto delle applicazioni all'oggetto code. Invece, il programma di gestione code funziona da intermediario tra i programmi di applicazione e le code. Le code si distinguono in base al nome, che può essere una stringa di caratteri ASCII di lunghezza illimitata<sup>3</sup> ma deve essere unico all'interno di uno specifico programma di gestione code. MQSeries Everyplace supporta un numero di tipi diversi di code:

### Code locali

Le code locali sono utilizzate dalle applicazioni per memorizzare i messaggi in modo protetto. Il messaggio memorizzato viene assegnato alla memoria permanente attraverso un adattatore, su una base "da coda a coda". Con MQSeries Everyplace viene fornita una serie di adattatori; altri possono essere scritti o sono disponibili da altre fonti. L'adattatore standard è noto come MQeDiskFieldsAdapter e associa le code nel file system locale e implementa la consegna assicurata. Un altro adattatore, noto come MQeReducedDiskFieldsAdapter, associa le code nel file system, ma scambia l'esecuzione più veloce per

---

3. Per l'interoperabilità si consiglia di rispettare i limiti di denominazione di MQSeries, compreso un nome di lunghezza massima di 48 caratteri. La lunghezza può essere limitata dal file system utilizzato

dipendere dal sistema operativo abbastanza duraturo da svuotare i propri buffer nel sistema secondario del disco. Invece, l'adattatore MQMemoryFieldsAdapter associa le code nella memoria. Nel fornire l'esecuzione più veloce della coda, tale adattatore dispone della caratteristica per la quale i messaggi non si conservano al riavvio del sistema operativo o del programma di gestione code. Creano l'adattatore più appropriato, è possibile memorizzare i messaggi ovunque, su una base "da coda a coda". Alcuni esempi di supporti di memorizzazione sono un database relazionale o un CD scrivibile. Gli adattatori sfruttano il file system speculare sul controller IBM 4690 retail store oppure il DB2 per la memoria delle code.

È possibile utilizzare le code locali in linea o non in linea (oppure connesse o meno ad una rete). Le code possono anche disporre di un insieme di attributi di sicurezza, in modo simile alla protezione degli oggetti messaggi e campi con l'oggetto attributi. La sicurezza delle code viene trattata in "Sicurezza" a pagina 48. L'accesso ai messaggi sulle code locali è sempre sincrono; ciò significa che l'applicazione attende fino a quando MQSeries Everyplace non torna dopo aver completato l'operazione di inserimento o di ricezione.

### Code remote

Le code remote sono riferimenti locali alle code che si trovano in un programma di gestione code remoto. Il riferimento locale dispone dello stesso nome della coda di destinazione ma la definizione della coda remota identifica il programma corrente di gestione code della coda reale. Le code remote dispongono, inoltre, di proprietà relative all'accesso, come, ad esempio, la modalità di accesso (sincrona o asincrona), le caratteristiche di sicurezza e le opzioni di trasmissione.

MQSeries Everyplace può stabilire automaticamente le code remote. Se viene effettuato un tentativo di accesso (per esempio per inviare un messaggio) a una coda in un altro programma di gestione code, MQSeries Everyplace cerca una definizione della coda remota. Ne viene utilizzata una qualora presente, in caso contrario si verifica la *scoperta della coda*. Vengono scoperte le caratteristiche (autenticazione, codifica e compressione) e viene creata una definizione della coda remota. Tale scoperta della coda dipende dalla destinazione accessibile. Se la destinazione non è accessibile, è necessario fornire in qualche modo una definizione remota. Quando si verifica la scoperta della coda, MQSeries Everyplace imposta l'accesso in modalità sincrona poiché la coda è disponibile sincronicamente.

Le code remote *sincrone* sono code a cui è possibile accedere solo quando si è collegati a una rete dotata di un percorso delle comunicazioni per il programma di gestione code in questione. Se non viene stabilito il collegamento alla rete, operazioni come l'inserimento

(put), la ricezione (get) e lo sfoglio (browse) (vedere “Operazioni dei programmi di gestione code” a pagina 38), provocano un’eccezione. La coda corrente controlla i permessi di accesso e i requisiti di sicurezza necessari per l’accesso alla coda. È responsabilità dell’applicazione gestire eventuali errori o nuovi tentativi per l’invio o la ricezione di messaggi poiché, in questo caso, MQSeries Everyplace non sarà più responsabile per la consegna unica assicurata.

Le code remote *asincrone* sono code che inviano messaggi alle code remote ma non possono recuperarli in remoto. Se viene stabilito il collegamento alla rete i messaggi vengono inviati alla coda e al programma di gestione code corrente. Se, tuttavia, la rete non è collegata, i messaggi vengono memorizzati in locale fino al momento di un collegamento a una rete e quando i messaggi verranno trasmessi. Ciò consente alle applicazioni di operare sulla coda quando il dispositivo non è in linea. Di conseguenza, queste code dispongono di un adattatore che crea delle corrispondenze nella memoria di un messaggio in modo tale che i messaggi possano essere memorizzati temporaneamente nel programma di invio di gestione code in attesa della trasmissione.

### Code di memorizzazione e inoltra

Questo tipo di coda memorizza i messaggi associati con una o più destinazioni dei programmi di gestione code. Sono possibili due tipi di utilizzo. Il primo abilita la memoria intermedia dei messaggi in una rete in modo che possano procedere passo dopo passo fino alla destinazione (una regola di inoltra). Il secondo utilizzo serve per conservare i messaggi in attesa di raccolta (vedere anche le code dell’home server).

Le code di memorizzazione e inoltra sono associate ad un insieme di nomi di programmi di gestione code per i quali conservano i messaggi. Ci si riferisce a questi come programmi di gestione code di destinazione. I messaggi indirizzati ad uno di questi programmi di destinazione verranno, invece, posizionati nella coda di memorizzazione e inoltra rilevante. La coda di memorizzazione e inoltra può, inoltre, disporre facoltativamente di un insieme di nomi del programma di gestione code di inoltra. Se questo nome viene impostato, la coda tenta di inviare tutti i messaggi a quel programma di gestione code a cui è stato assegnato un nome. Se il nome non viene impostato, la coda conserva soltanto i messaggi.

Questo tipo di coda viene definito in genere (ma non necessariamente) su un server o un gateway. Le code di memorizzazione e inoltra multiple possono trovarsi su un solo programma di gestione code, ma i nomi di destinazione non devono essere duplicati. I dati di una coda di memorizzazione e inoltra non sono disponibili per i programmi di

applicazione. Allo stesso modo, un messaggio che invia un'applicazione non considera la presenza e il ruolo delle code di memorizzazione e inoltrò nella trasmissione dei messaggi.

I messaggi sulle code di memorizzazione e inoltrò non sono disponibili per le applicazioni.

### Code dell'home server

Le code remote e le code di memorizzazione e inoltrò inviano messaggi attraverso la rete e le code di invio che avviano la trasmissione. Le code dell'home server consentono, comunque, di estrarre i messaggi da una coda remota. Una definizione della coda dell'home server identifica una coda di memorizzazione e inoltrò in un programma remoto di gestione code. La coda dell'home server estrae tutti i messaggi destinati al programma di gestione code locale della coda dell'home server dalla coda di memorizzazione e inoltrò. Le definizioni multiple della coda dell'home server possono essere stabilite su un solo programma di gestione code, in cui ognuna viene associata a una diversa coda remota di memorizzazione e inoltrò.

Le code dell'home server si trovano in genere su un dispositivo e vengono impostate per estrarre i messaggi da un server ogni volta che il dispositivo si connette alla rete. Quando una coda dell'home server estrae un messaggio dal server, il messaggio viene posizionato nella corretta coda locale di destinazione. In questo modo la coda dell'home server non dispone di messaggi accessibili alle applicazioni. Questo metodo di estrazione dei messaggi dal server può essere più efficiente in termini di flusso sulla rete rispetto al server che invia i messaggi. Questo dipende dal fatto che la coda dell'home server utilizza l'avviso di ricezione del primo messaggio come richiesta per il messaggio successivo (se presente), mentre il server per l'invio manda una richiesta/risposta per inviare il messaggio e una seconda richiesta/risposta per il flusso di conferma. Una coda dell'home server che in genere dispongono di un intervallo di poll impostato controlla tutti i messaggi in sospeso sul server mentre si è collegati alla rete. Questo intervallo di poll è un'opzione per la configurazione dell'amministrazione. Le code dell'home server ricoprono un ruolo importante nell'abilitare i client a ricevere i messaggi sui canali client-server. La natura del collegamento client-server implica che i server non possono avviare il trasferimento dei dati. <sup>4</sup>

I messaggi sulle code dell'home server non sono disponibili per le applicazioni.

---

4. L'alternativa è l'utilizzo dei canali peer-to-peer o la configurazione dei programmi di gestione code sia con le funzioni del client che del server.

## Code di amministrazione

Attraverso le code di amministrazione vengono configurati, o in locale o in remoto, i programmi di gestione code e gli oggetti ad essi associati. Un messaggio inviato alla coda di amministrazione viene elaborato dalla classe del messaggio di amministrazione rilevante e, quindi, viene rinviata, facoltativamente, una risposta all'applicazione di origine. Tale argomento viene trattato più dettagliatamente in "Amministrazione" a pagina 41.

## code bridge di MQSeries

Questa è una forma specializzata di coda remota con la definizione di un gateway e la coda di destinazione di un programma di gestione code di MQSeries. Questo tipo di coda fornisce un percorso tra gli ambienti di MQSeries Everyplace e MQSeries. I trasformatori vengono utilizzati per elaborare tutti i dati necessari o per la riformattazione dei messaggi. Un trasformatore di base viene fornito con MQSeries Everyplace; i programmatori devono, poi, personalizzare questo trasformatore per adeguarlo alle proprie necessità.

MQSeries Everyplace memorizza i dati sulle code in modo sicuro, assicurando, in base all'adattatore, che i messaggi vengano scritti fisicamente nei supporti e non semplicemente memorizzati nel buffer dal sistema operativo. Tuttavia MQSeries Everyplace non registra in modo indipendente le modifiche ai messaggi e alle code. Se è richiesto un recupero per un errore dei supporti, sarà necessario adottare delle soluzioni di hardware come l'utilizzo di sistemi di dischi RAID. Altrimenti la coda deve avere un corrispondente nella memoria recuperabile, per esempio alcuni sistemi secondari di database.

MQSeries Everyplace non richiede che il programma di gestione code definisca le code. Viene comunque preso un provvedimento per quattro code di sistemi, se richiesto:

- **AdminQ:** richiesto per la ricezione dei messaggi di amministrazione
- **AdminReplyQ:** utilizzato in modo facoltativo per le risposte ai messaggi di amministrazione
- **DeadLetterQ:** utilizzato per memorizzare i messaggi che non è possibile recapitare in altro modo
- **SYSTEM.DEFAULT.LOCAL.QUEUE:** una coda che condivide un nome uguale con la coda di sistema obbligatoria sui server di MQSeries

Le proprietà delle code sono illustrate nella seguente tabella. Tuttavia, non tutte le proprietà visualizzate sono valide per tutti i tipi di code:

*Tabella 8. Proprietà delle code*

Proprietà	Spiegazione
Admin_Class	Classe di code

Tabella 8. Proprietà delle code (Continua)

Proprietà	Spiegazione
Admin_Name	Nome coda Ascii
Queue_Active	Indica che la coda è attiva
Queue_AttRule	Classe di regole che controllano le operazioni di sicurezza
Queue_Authenticator	Classe autenticatore
Queue_BridgeName	Appartenenza nome MQSeries-bridge
Queue_ClientConnection	Nome collegamento al client
Queue_CloseIdle	Chiude il trasportatore una volta trasmessi tutti i messaggi
Queue_CreationDate	La data in cui la coda è stata creata
Queue_Compressor	Classe compressore
Queue_Cryptor	Classe codificatore
Queue_CurrentSize	Numero di messaggi sulla coda
Queue_Description	Descrizione di UNICODE
Queue_Expiry	Scadenza dei messaggi
Queue_FileDesc	Posizione e adattatore per la coda
Queue_MaxMsgSize	Lunghezza massima dei messaggi consentiti sulla coda
Queue_MaxQSize	Num. massimo di messaggi consentiti
Queue_Mode	Sincrono o asincrono
Queue_MQMgr	proxy del programma di gestione code di MQSeries
Queue_Priority	Priorità per i messaggi (se non sovrascritta da un valore di messaggio)
Queue_QAliasNameList	Nomi alternativi per la coda
Queue_QMgrName	Programma di gestione code che gestisce la coda reale
Queue_QMgrNameList	Destinazioni dei programmi di gestione code
Queue_RemoteQName	Nome campo remoto di MQSeries
Queue_Rule	Classe di regole per le operazioni sulle code
Queue_QTimerInterval	Ritardo prima dell'elaborazione dei messaggi in attesa
Queue_TargetRegistry	Il tipo di registro di destinazione
Queue_Transporter	Classe trasportatore

Tabella 8. Proprietà delle code (Continua)

Proprietà	Spiegazione
Queue_TransporterXOR	Trasportatore per utilizzare la compressione XOR
Queue_Transformer	Classe trasformatore

Le funzioni amministrative consentono di creare ed eliminare le code oltre che ricercare o modificare le proprietà.

I seguenti pannelli di MQE\_Explorer mostrano due delle quattro schede che descrivono le proprietà di una coda locale. I campi disabilitati indicano che una particolare proprietà non è rilevante per una coda di quella classe.

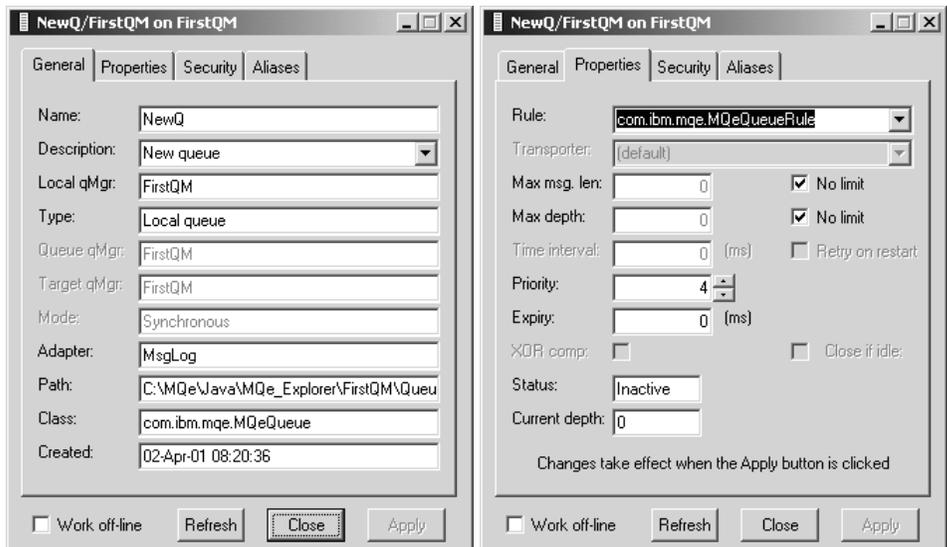


Figura 5. Proprietà della coda locale visualizzata conMQE\_Explorer

Le code non sono limitate solo all'utilizzo come destinazione per i messaggi. Le code di classe secondaria sono utilizzate negli scenari per le applicazioni di controllo processi, ad esempio l'oggetto coda può controllare direttamente un valore. Un messaggio della classe corretta può far aprire la valvola, modificare il volume del flusso e via di seguito. Un'applicazione non sposterà i messaggi fuori dalla coda per eseguire l'operazione, l'oggetto coda stesso controlla l'operazione. Altre code possono, ad esempio, aggiornare fogli elettronici o eseguire delle conversioni da testo in dialogo. I vantaggi di questa tecnica consistono nel fatto che gli aspetti della sicurezza delle code sono sempre operativi, così come viene garantita la messaggistica. In questo modo MQSeries Everyplace assicura la consegna unica dei messaggi, mentre un programma di autenticazione e uno di codifica associati garantiranno che solo

il mittente autorizzato del messaggio potrà inviare tali messaggi, con contenuto altamente sicuro nel trasferimento. A nessuna applicazione è consentito l'accesso alla coda, né verrà richiesto.

---

## Programmi di gestione code

Il programma di gestione code di MQSeries Everyplace fornisce l'accesso delle applicazioni a messaggi e code, e controlla tutti i canali. In MQSeries Everyplace Versione 1.2 solo un programma di gestione code può essere attivo su ogni singola macchina virtuale Java in qualsiasi momento. Se esistono più JVM su un computer, può essere presente lo stesso numero di programmi di gestione code delle JVM. I programmi di gestione code vengono identificati per nome e questo deve essere globalmente unico<sup>5</sup> Una stringa di caratteri Ascii può essere di lunghezza illimitata.<sup>6</sup>

Un gestore code configurato con funzioni del bridge e, pertanto, in grado di scambiare i messaggi con i prodotti distribuiti e per host di MQSeries viene definito gateway.

È possibile configurare i gestori code con o senza l'accodamento locale. Tutti i programmi di gestione code supportano operazioni di messaggistica sincrona; un programma di gestione code con accodamento locale supporta anche la consegna sincrona dei messaggi.

La scelta tra l'utilizzo della consegna sincrona o asincrona del messaggio dipende dalla natura delle definizioni della coda nel programma di gestione code di invio. Se esiste per la coda di destinazione una definizione della coda remota in modo sincrono, verrà utilizzata la consegna sincrona. In caso contrario, viene utilizzata la consegna asincrona, con la definizione che fornisce la memoria locale mentre i messaggi attendono la trasmissione. Viene utilizzata la consegna asincrona nel caso in cui non ci sia alcuna definizione della coda remota ma sia presente una coda di memorizzazione e inoltre che gestisce i messaggi per quel programma di gestione code di destinazione. In questo caso, la coda di memorizzazione e inoltre fornisce la memoria locale per i messaggi in attesa di trasmissione. Se non esistono definizioni di code, ha luogo la scoperta della coda; se questa avviene correttamente, ne deriva la messaggistica sincrona.

Senza prendere in considerazione se viene utilizzata la messaggistica sincrona o asincrona, MQSeries Everyplace può utilizzare la trasmissione diretta o

---

5. Questa limitazione non è applicata da MQSeries Everyplace o MQSeries, ma nomi di programmi di gestione duplicati possono far recapitare messaggi al programma di gestione sbagliato.

6. Per l'interoperabilità si consiglia di rispettare le regole di MQSeries per la denominazione dei programmi di gestione code, compreso un nome di lunghezza massima di 48 caratteri. La lunghezza, inoltre, può essere limitata dal file system utilizzato.

## programmi di gestione code

indiretta, a seconda delle definizioni di connessione disponibili (vedere “Connessioni” a pagina 39). La trasmissione diretta implica due programmi di gestione code, quello di invio e quello di destinazione. La connessione indiretta implica una successione di programmi di gestione code, con possibili modifiche di protocollo in viaggio. Quando viene utilizzata la trasmissione indiretta con la messaggistica asincrona, il funzionamento di quest’ultima rimane invariato. I programmi di gestione code intermedi stabiliscono semplicemente le connessioni tra l’origine e la destinazione. Tuttavia, quando viene utilizzata la trasmissione asincrona con la messaggistica asincrona, gli intermediari di trasmissione possono diventare messaggi di gestione temporanea per il messaggio quando si sposta dall’origine alla destinazione. Vengono utilizzati in questo modo se la memoria della coda intermedia è stata definita nei relativi programmi di gestione code, nella forma di appropriate definizioni di code remote o di code di memorizzazione e inoltro.

La consegna sincrona e asincrona di messaggi presenta caratteristiche e conseguenze molto diverse. Con la consegna asincrona dei messaggi, l’applicazione passa il messaggio a MQSeries Everyplace per recapitarlo ad una coda remota. Si ritorna immediatamente all’applicazione. Se è possibile consegnarlo immediatamente o spostarlo in un messaggio di gestione temporanea adatto, il messaggio viene inviato; in caso contrario, viene memorizzato in locale. La logica dei tentativi di trasmissione viene definita dalle regole associate al programma di gestione code e alle code rilevanti (vedere “Regole” a pagina 56). La consegna asincrona fornisce la qualità del servizio di consegna assicurata una sola volta, poiché il messaggio è stato inviato a MQSeries Everyplace ed è diventato responsabile per la consegna.

Con la consegna asincrona dei messaggi, l’applicazione passa il messaggio a MQSeries Everyplace per recapitarlo alla coda remota. MQSeries Everyplace contatta in modo sincrono la coda di destinazione e posiziona il messaggio. Dopo la consegna MQSeries Everyplace ritorna all’applicazione. Si ritorna immediatamente all’applicazione. Se il messaggio non può essere consegnato, l’applicazione di invio riceve una notifica immediata. MQSeries non si assume responsabilità per la consegna del messaggio in modo sincrono.

Le definizioni delle code remote sincrone e asincrone possono essere liberamente stabilite su una rete di MQSeries Everyplace. Quando la rete comprende anche MQSeries che invia messaggi ai programmi di gestione code e alle relative code, diventano importanti alcune limitazioni.

1. La messaggistica sincrona non può essere effettuata in un programma di gestione code di MQSeries non direttamente collegato ad un gateway di MQSeries Everyplace, poiché la messaggistica sincrona non è supportata sui canali dei messaggi di MQSeries. Allo scopo di ridurre le conseguenze, la definizione della consegna sincrona viene in questo caso modificata e ridefinita in quanto in consegna al programma di gestione code di MQSeries, direttamente collegato al gateway. Oltre a questo programma di

gestione code, la messaggistica asincrona di MQSeries viene utilizzata per consentire al messaggio di completare il percorso.

2. In MQSeries Everyplace Versione 1.2 è possibile stabilire soltanto una definizione della coda remota sincrona ad un MQSeries che invia messaggi alla coda remota. Tenendo conto dell'impatto di quanto riportato sopra (1), ciò significa che, utilizzando questa definizione, la consegna asincrona non può essere effettuata nelle code su un programma di gestione code di MQSeries, direttamente collegato al gateway. Tuttavia, se in questo caso viene richiesto l'accodamento, è possibile organizzarlo nella rete di MQSeries Everyplace utilizzando un secondo programma di gestione code di MQSeries Everyplace, attraverso l'impiego di appropriate definizioni delle code remote o delle code di memorizzazione e inoltre.

La consegna asincrona dei messaggi implica che l'applicazione locale consegna il messaggio a MQSeries Everyplace; da questo punto in poi la consegna dal programma di gestione code locale è responsabilità di MQSeries Everyplace. Quindi non è necessario che la rete e/o l'applicazione per la ricezione siano disponibili. L'ora dell'effettiva consegna è sconosciuta all'applicazione per l'invio. Per la consegna sincrona dei messaggi è necessario che la rete sia attiva, ma l'applicazione per l'invio saprà che il messaggio viene recapitato alla coda dell'applicazione per la ricezione. L'applicazione per la ricezione non deve necessariamente essere disponibile, sia nel caso della consegna asincrona che sincrona.

MQSeries Everyplace non offre variazioni sulla qualità dei servizi di consegna assicurata per una volta sola per i messaggi asincroni come avviene per i messaggi di MQSeries fornito di opzioni permanenti e non. In MQSeries Everyplace gli scambi tra l'affidabilità e le prestazioni possono essere effettuati al livello di code scegliendo un adattatore coda di memorizzazione, ad esempio:

- MQeDiskFieldsAdapter garantisce la scrittura accurata dei dati su disco prima che il processo continui
- MQeReducedDiskFieldsAdapter garantisce il passaggio dei dati al sistema operativo prima che il processo continui
- MQeMemoryFieldsAdapter salva i dati in memoria

In tutti i casi, viene effettuata la consegna assicurata per una volta sola e comunque, il ripristino dipende dalla memoria del messaggio scelto. Gli scambi di esecuzione vengono descritti in SupportPac EP01.

Le proprietà dei programmi di gestione code vengono indicate in Tabella 9 a pagina 34.

## programmi di gestione code

Tabella 9. Proprietà dei programmi di gestione code locali

Proprietà	Descrizione
Admin_Class	Classe del programma di gestione code
Admin_Name	Nome del programma di gestione code
QMgr_ChnlAttrRules	Regole di attributi di canali
QMgr_ChnlTimeout	Time-out del canale
QMgr_Description	descrizione di UNICODE
QMgr_QueueStore	Posizione predefinita e l'adattatore per le code
QMgr_Rules	Classe di regole per le operazioni dei programmi di gestione code

Il seguente pannello mostra una delle due schede di proprietà che descrive le proprietà di un programma di gestione code:

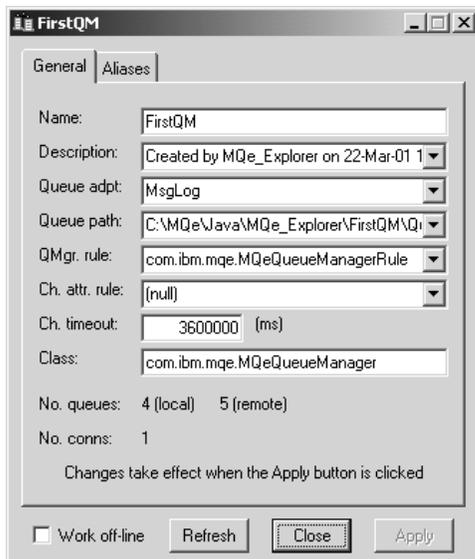


Figura 6. Proprietà del programma di gestione code visualizzato da MQE\_Explorer

### Configurazione del programma di gestione code

Tale programma viene eseguito in un ambiente stabilito da MQSeries Everyplace prima del suo caricamento. Lo stesso programma, memorizza le informazioni relative alla configurazione nel proprio registro (descritto più dettagliatamente in "Il registro" a pagina 53). Le stesse code (contenenti messaggi) vengono memorizzate nelle memorie di coda.

L'ambiente di MQSeries Everyplace può essere stabilito in diversi modi, sia attraverso le chiamate effettuate con API, sia attraverso i programmi di utilità forniti con MQSeries Everyplace oppure attraverso gli strumenti di gestione come MQe\_Explorer. La maggior parte degli strumenti catturano i parametri di ambiente in un file di inizializzazione, sebbene sia un'opzione. La seguente tabella mostra i parametri di ambiente; i nomi di sezione utilizzati seguono le convenzioni impiegate per indicare le informazioni nei file di inizializzazione:

*Tabella 10.*

Nome della sezione	Proprietà	Spiegazione
[Alias]	(definizioni di alias di classe)	I nomi alias possono essere utilizzati qualora vengano richiesti i nomi classe
[ChannelManager]	MaxChannels	Numero massimo di canali client-server simultanei consentiti
[Listener]	Rete	L'adattatore e il numero di porta per ricevere le richieste relative alla connessione in arrivo del canale client/server
	Listen	L'adattatore da utilizzare per gestire le connessioni risultanti
	TimeInterval	Il time-out del canale client/server
[MQBridge]	(parametri di inizializzazione del bridge)	
[MQe_Explorer]	(informazioni dell'indirizzo salvato)	MQe_Explorer salva le informazioni relative alle connessioni in arrivo effettuate in questo programma di gestione code
[Permission]	(comandi permessi)	Comandi di canale, classe di adattatori e corrispondenze di descrittori file permessi
[PreLoad]	classi da caricare quando viene inizializzato il programma di gestione code	Ciò consente il caricamento delle classi di applicazione
[QueueManager]	Nome	Nome del programma di gestione code

## programmi di gestione code

Tabella 10. (Continua)

Nome della sezione	Proprietà	Spiegazione
[Registry]	DirName	Posizione del registro
	LocalRegType	Il tipo di registro (di file o privato)
	PIN	PIN del registro (o alla richiesta)
	CertReqPIN	PIN del certificato (o alla richiesta)
	KeyRingPassword	Password Key ring (o alla richiesta)
	CAIPAddrPort	Indirizzo IP dell'autorità di certificazione

Le seguenti quattro illustrazioni mostrano le schede chiave in grado di creare un nuovo programma di gestione code utilizzando MQe\_Explorer. Non viene supposto alcun ambiente e non viene richiesto alcun file di inizializzazione esistente. Il risultato è il programma di gestione code in esecuzione, i cui dati relativi alla configurazione sono salvati su un file di inizializzazione; il programma, quindi, può essere riavviato semplicemente aprendo questo file.

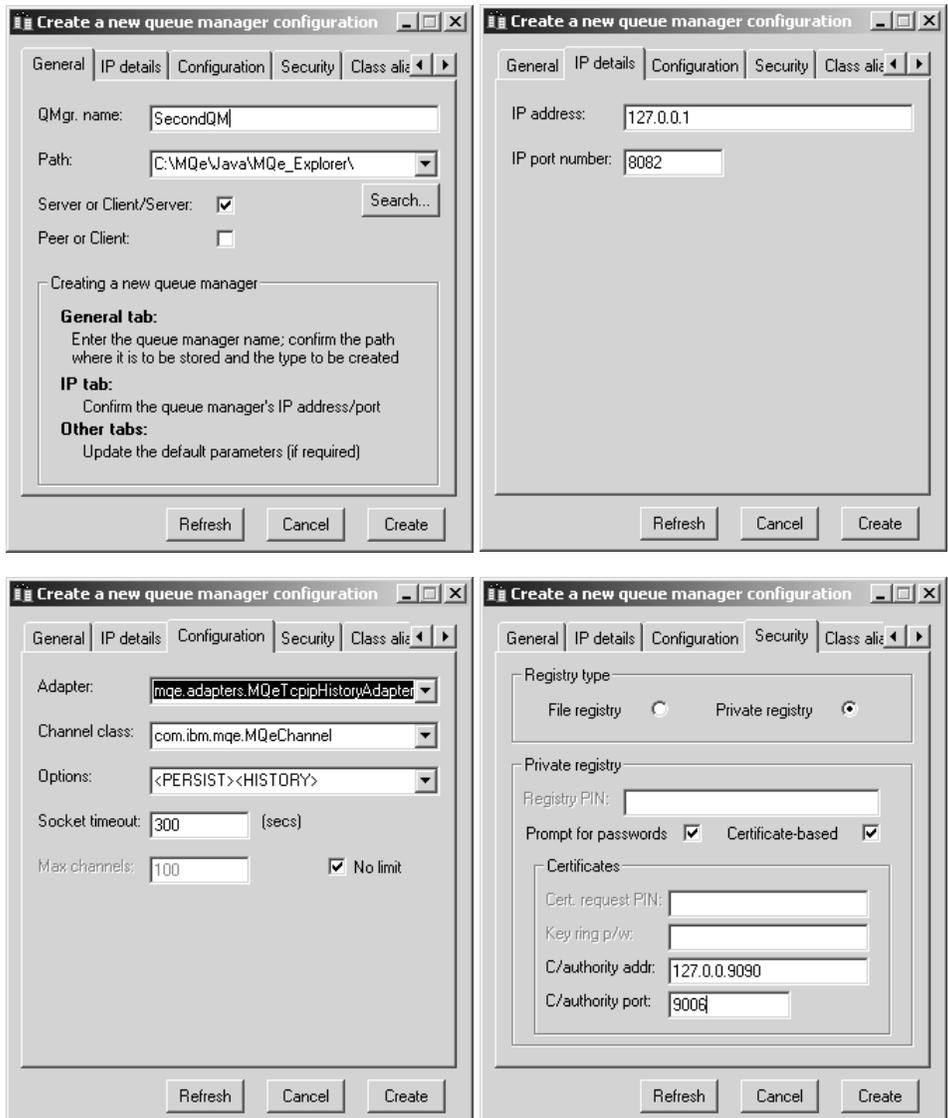


Figura 7. Creazione di un programma di gestione code utilizzando MQe\_Explorer

Non è necessario conoscere l'indirizzo IP del programma di gestione code per crearlo. Tuttavia, MQe\_Explorer acquisisce le informazioni relative alla configurazione di altri programmi di gestione code. Viene richiesto il numero della porta per attendere le richieste relative alla connessione in arrivo client/server.

I dati di configurazione dell'adattatore sono necessari al listener di canali; l'intervallo di tempo e il numero massimo di canali sono necessari al gestore

## programmi di gestione code

di canali. Come precedentemente illustrato, MQE\_Explorer acquisisce tutte le altre informazioni, per permettere la configurazione degli altri programmi di gestione code.

In questo esempio, un registro protetto viene configurato in base al programma di autenticazione basato su certificato (consultare “Sicurezza” a pagina 48). I PIN e le password sono disabilitati poiché MQE\_Explorer potrebbe richiederle solo successivamente.

### Operazioni dei programmi di gestione code

I programmi di gestione code supportano le operazioni di messaggistica e in modo facoltativo gestiscono le code. Le applicazioni accedono ai messaggi attraverso i servizi del programma di gestione code utilizzando alcuni metodi quali *ricevere (get)*, *inserire (put)*, *sfogliare (browse)*, *attendere (wait)*, *attendere (listen)* e *eliminare (delete)*. Molte di queste operazioni hanno come parametro un *filtro*. Un filtro è un oggetto campo di cui si trova una corrispondenza per uguaglianza e tutti i campi del messaggio possono essere utilizzati per un recupero selettivo. La maggior parte dei metodi di ricezione includono, inoltre, un oggetto attributo da utilizzare per la codifica e la decodifica di un messaggio.

L'operazione *get* rimuove i messaggi da una coda danneggiandoli. In base alle condizioni imposte dal filtro, i messaggi vengono recuperati secondo un ordine di priorità e poi, in ordine di tempo di arrivo sulla coda. Per cui, a parità di condizioni, il primo messaggio arrivato sarà quello che per primo verrà recuperato. *Get* sarà disponibile come operazione a fase singola o doppia. La fase doppia viene fornita nei casi in cui è essenziale che non venga perso alcun messaggio, poiché i messaggi passano direttamente da MQSeries Everyplace all'applicazione. Prima viene emesso un *get ID di conferma* (con valore scelto dall'applicazione). Questa operazione riceve il messaggio ma lo nasconde sulla coda invece di eliminarlo immediatamente. Una successiva conferma, che specifica l'UID di messaggio originale, indicherà che la ricezione per l'applicazione è avvenuta correttamente; a questo punto il messaggio verrà eliminato. Se la ricezione (*get*) non avviene correttamente, sarà possibile recuperare il messaggio. Le operazioni di invio (*put*) si svolgono in modo simile.

Specificando l'UID, i messaggi possono essere *eliminati* da una coda senza poter essere recuperati.

Se viene richiesta una lettura non distruttiva, è possibile *ricercare* le code per i messaggi (se si desidera, sotto il controllo di un filtro). Con la ricerca vengono recuperati tutti gli oggetti messaggi che corrispondono al filtro, ma vengono lasciati sulla coda. È supportata anche la *ricerca con blocco*. Esiste una funzione aggiuntiva per bloccare i messaggi corrispondenti sulla coda. I messaggi possono essere bloccati singolarmente, oppure in gruppi identificativi da un

filtro, e l'operazione di blocco darà come risultato un *ID blocco*. I messaggi bloccati possono essere ricevuti o eliminati solo se si fornisce l'ID blocco. Un'opzione di ricerca consente di recuperare l'intero messaggio o solo l'UID.

Le applicazioni possono *attendere* un periodo di tempo specificato l'arrivo dei messaggi su una coda. Oppure è possibile utilizzare un filtro per identificare i messaggi pertinenti e specificare un *ID di conferma*. Le applicazioni possono infine attendere gli eventi messaggio di MQSeries Everyplace, in modo facoltativo con un filtro. L'utente in attesa verrà avvisato all'arrivo dei messaggi su una coda.

Le code sono abilitate per operazioni di messaggistica come indicato in Tabella 11

Tabella 11. Operazioni di messaggistica su code MQSeries Everyplace

	Coda locale	Coda remota <sup>1</sup>	
		Sincrono	Asincrono
Sfoggia (±blocco, ±filtro)	Sì	Sì	
Elimina	Sì	Sì	
Get (±filtro)	Sì	Sì	
Attendi (±filtro)	Sì		
Put	Sì	Sì	Sì
Attendi (±filtro)	Sì	Sì	

**Note:**

1. L'operazione di attesa remota sincrona è implementata con un poll della coda remota, quindi il tempo di attesa attuale è un multiplo del tempo di poll
2. <sup>1</sup>Il Bridge MQSeries Everyplace MQSeries fornito con MQSeries Everyplace Versione 1.2 supporta solo l'operazione put sicura/incerta, get incerta, e browse incerta (senza blocco).

## Connessioni

La tipologia e l'accesso attraverso la rete di MQSeries Everyplace vengono definiti dagli oggetti di collegamento. Queste definizioni vengono localmente memorizzate in ogni programma di gestione code. Vengono create, modificate ed annullate in base all'amministrazione standard.

Un oggetto di collegamento, generalmente definisce l'accesso ad un programma di gestione code remoto (spesso ci si riferisce a definizioni del programma di gestione code remoto). La seguente tabella illustra le proprietà:

Tabella 12. Proprietà di collegamento (programma di gestione code remote)

Proprietà	Spiegazione
Admin_Name	Nome del programma di gestione code
Con_Adapter	Il programma di descrizione file dell'adattatore
Con_AdapterOptions	Opzioni dell'adattatore (per esempio la cronologia di utilizzo)
Con_AdapterParm	Dati ASCII da utilizzare con un adattatore (per esempio un nome di servlet)
Con_Aliases	Nomi alternativi per i gestori code/collegamenti
Con_Channel	Il tipo di canale che il collegamento deve utilizzare
Con_Description	descrizione di UNICODE
Queue_QMgrName	Proprietario della definizione

I seguenti pannelli di amministrazione mostrano due dei separatori associati alla definizione di collegamento:

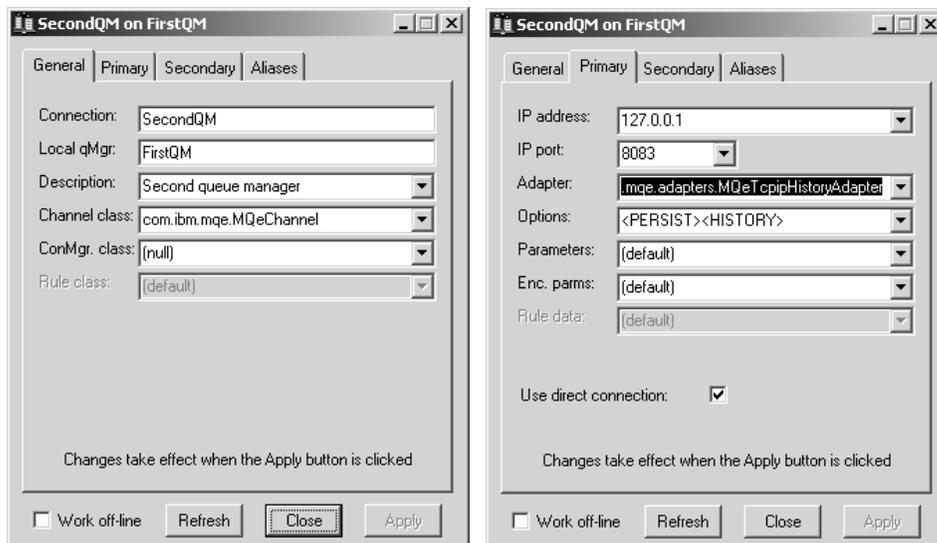


Figura 8. Definizione di collegamento visualizzata con MQE\_Explorer

I dati possono essere passati all'adattatore di comunicazioni scelto in qualità di opzioni, parametri e parametri codificati.

I collegamenti indiretti possono essere specificati. In questo caso, MQSeries Everyplace invia il collegamento attraverso altri programmi di gestione code

(che possono essere concatenati), ed il protocollo può essere modificato in viaggio. I collegamenti indiretti sono particolarmente utili per far in modo che i dispositivi abbiano un solo punto di collegamento ad una rete di MQSeries Everyplace.

Gli oggetti di collegamento vengono inoltre utilizzati per definire i listener dei canali in entrata peer-to-peer.

Come per la maggior parte degli oggetti di MQSeries, gli alias possono essere definiti dai collegamenti. Un collegamento locale, definito come collegamento il cui nome viene associato a quello del programma di gestione code locale, viene utilizzato per definire i nomi alias dello stesso programma di gestione code locale.

---

## Amministrazione

L'amministrazione fornisce delle funzioni per configurare e gestire risorse di MQSeries Everyplace come code e collegamenti. Le funzioni relative ai messaggi sono considerate responsabilità delle applicazioni.

L'amministrazione viene abilitata tramite un'interfaccia che gestisce la creazione e la ricezione di messaggi di amministrazione ed è progettata per gestire amministrazione locale e remota allo stesso modo. Le richieste vengono inviate alla coda di amministrazione del programma di gestione code prescelto e, se richiesto, è possibile ricevere risposte. Tutti i programmi di applicazione locali o remoti di MQSeries Everyplace possono creare ed eseguire messaggi di amministrazione direttamente o indirettamente tramite metodi di aiuto. I messaggi di amministrazione possono essere creati indirettamente con MQe\_Explorer<sup>7</sup>, uno strumento di gestione che fornisce un'interfaccia utente grafica per l'amministrazione dei sistemi.

La coda di amministrazione non capisce in che modo svolgere l'amministrazione di risorse individuali: questo metodo viene incapsulato in ciascuna risorsa e nel messaggio di amministrazione corrispondente.

### Messaggi di amministrazione

I messaggi di amministrazione ampliano l'oggetto messaggio base di MQSeries Everyplace. Tabella 13 a pagina 42 elenca le classi di messaggio fornite per l'amministrazione di risorse MQSeries Everyplace. Questi messaggi di amministrazione di base possono diventare classi secondarie ed essere utilizzati per l'amministrazione di altri oggetti; ad esempio un tipo diverso di coda può essere gestito utilizzando una classe secondaria di MQeQueueAdminMsg. Il bridge di MQSeries Everyplace a MQSeries utilizza classi secondarie di MQeAdminMsg in questo modo.

---

7. MQe\_Explorer non incluso in MQe\_Explorer ma sarà possibile scaricarlo dal sito Web di MQSeries Everyplace all'indirizzo (<http://www.ibm.com/software/ts/mqseries/everyplace>).

Tabella 13. Classi di messaggi di amministrazione

Classe di messaggi di amministrazione	Utilizzo
MQeAdminMsg	Classe astratta utilizzata come base per tutti i messaggi di amministrazione
MQeQueueManagerAdminMsg	Amministrazione dei programmi di gestione code
MQeQueueAdminMsg	Amministrazione delle code locali
MQeRemoteQueueAdminMsg	Amministrazione delle code remote
MQeAdminQueueAdminMsg	Amministrazione della coda di amministrazione
MQeHomeServerQueueAdminMsg	Amministrazione delle code dell'home server
MQeStoreAndForwardQueueAdminMsg	Amministrazione delle code di memorizzazione e inoltra
MQeConnectionAdminMsg	Gestione delle connessioni tra i programmi di gestione code
MQeClientConnectionAdminMsg	Amministrazione di un oggetto di connessione al client del bridge, utilizzato per la connessione a MQSeries
MQeListenerAdminMsg	Amministrazione di un oggetto di attesa della coda di trasmissione per il bridge, utilizzato per raccogliere messaggi da MQSeries
MQeBridgeAdminMsg	Amministrazione di un bridge a MQSeries
MQeMQBridgesAdminMsg	Amministrazione di elenco di MQSeries-bridge
MQeMQMgrProxyAdminMsg	Amministrazione di una rappresentazione del bridge di un gestore code MQSeries
MQeMQBridgeQueueAdminMsg	Amministrazione di una coda MQSeries-bridge

La struttura di un messaggio di amministrazione dipende dalla classe di appartenenza, quindi dalla natura della risorsa in gestione e dai dettagli dell'operazione da svolgere su questa risorsa. In genere comunque i messaggi di amministrazioni sono strutturati come indicato in Tabella 14 a pagina 43:

Tabella 14. Struttura generica di un messaggio di amministrazione

Campi livello 1	Campi livello 2 e successivi	Utilizzo
Admin_Action		Creare, eliminare, ricercare e così via
Admin_Errors		Oggetti campi principali
	Campi multipli	Informazioni dettagliate su base di errori
Admin_MaxAttempts		Numero massimo di volte in cui è possibile eseguire un'operazione di amministrazione
Admin_Parameters		Oggetti campi principali
	Risorsa	Nome della risorsa da gestire
	Campi multipli	Dati dettagliati di parametri specifici per classe di messaggio e operazione
Admin_Reason		Messaggio di testo che indica il motivo di un errore
Msg_ReplyToQ		Nome della coda a cui inviare la risposta al messaggio
Msg_ReplyToQMgr		Nome del programma di gestione code a cui inviare la risposta a un messaggio
Admin_RC		Codice numerico di errore che indica il risultato
Msg_Style		Comando o richiesta/risposta
Admin_TargetQMgr		Nome del programma di gestione code che si occupa della risorsa di destinazione

Sono supportati tre stili di messaggi di amministrazione, cioè comandi (datagrammi) che indicano un'operazione di amministrazione che non richiede risposte, richieste che richiedono una risposta, e infine le risposte

## amministrazione

stesse. La risposta viene creata da una copia del messaggio originale; è possibile poi il mittente aggiunga dei campi supplementari che il ricevente può utilizzare.

### Amministrazione selettiva

È possibile controllare l'accesso all'amministrazione attraverso lo strumento di autenticazione sulla coda di amministrazione. Lo strumento di autenticazione fornito considera le applicazioni locali per rappresentare lo stesso utente locale, quindi abilita o meno l'amministrazione per tutte. Le applicazioni di amministrazione remota sono controllate dal richiamo dello strumento di autenticazione sul canale prima di qualsiasi flusso di messaggi di amministrazione. Sarà possibile così distinguere i vari utenti remoti e abilitarli o meno in modo separato. In tutti i casi, per tutti gli utenti, l'amministrazione viene abilitata o meno in modo completo. Se è richiesto un modello più avanzato di controllo sull'amministrazione, per esempio alcuni utenti di amministrazione hanno accesso a determinate code mentre altri no, sarà richiesta una ulteriore programmazione. Uno strumento di autenticazione più sofisticato può mantenere la traccia delle autorizzazioni associate alle identità degli utenti, di conseguenza i messaggi di amministrazione possono essere elaborati sulla base di queste autorizzazioni (consultare la sezione "Sicurezza" a pagina 48). Le regole associate alle code vengono poi sfruttate per abilitare o meno le operazioni in modo simile (consultare "Regole" a pagina 56).

### Programma di controllo e operazioni relative

L'amministrazione va spesso oltre la creazione e la modifica degli oggetti. Può includere il controllo del sistema ed informare un operatore quando una coda è quasi piena o quali sono le operazioni da eseguire all'arrivo di un messaggio troppo grande. Questi aspetti sono gestiti in MQSeries Everyplace utilizzando delle regole, ovvero delle classi richiamate ogni volta che gli oggetti modificano in modo sostanziale il loro stato oppure quando si verificano determinate situazioni di errore. Con MQSeries Everyplace viene fornita una serie predefinita di classi di regole, ma in genere vengono sostituite con classi personalizzate (vedere "Regole" a pagina 56).

---

## Canali dinamici

MQSeries Everyplace comunica tra gestioni code attraverso collegamenti logici che vengono definiti canali dinamici. Questi supportano flussi bidirezionali e sono stabiliti dal programma di gestione code come richiesto. La messaggistica sincrona e asincrona utilizzano gli stessi canali e un protocollo unico per MQSeries Everyplace. Questo protocollo può essere personalizzato sulla base del messaggio, sostituendo il metodo di scaricamento del messaggio. Diversamente, MQSeries utilizza in genere dei canali client per il traffico di tipo sincrono e una coppia di canali di messaggi per la

messaggistica asincrona bidirezionale. I *canali di messaggi cluster* di MQSeries hanno caratteristiche simili ai canali dinamici di MQSeries Everyplace, anche se esistono alcune differenze.

Un canale dinamico è un collegamento logico tra due programmi di gestione code, che ha lo scopo di inviare e ricevere dati. Possono esistere più canali contemporaneamente, anche tra gli stessi elementi, con le seguenti caratteristiche: autenticazione, codifica, compressione e protocollo di trasporto utilizzato. Queste caratteristiche sono collegabili, (è possibile utilizzare diverse versioni su diversi canali) per cui ciascun canale ha una propria qualità di attributi di servizi:

- **Strumento di autenticazione:** un oggetto *strumento di autenticazione* o nullo per eseguire l'autenticazione di un utente o un canale
- **Canale:** la classe che fornisce servizi di trasporto.
- **Compressore:** un oggetto *compressore* o nullo in grado di eseguire la compressione e decompressione dei dati
- **Programma di codifica:** un oggetto *programma di codifica* o nullo in grado di eseguire codifica e decodifica
- **Destinazione:** la destinazione per il canale, ad esempio SERVER.XYZ.COM

Lo strumento di autenticazione si utilizza in genere solo per l'impostazione del canale. I compressori e i programmi di codifica sono invece utilizzati per tutti i flussi.

Il tipo più semplice di programma di codifica è MQeXorCryptor, che codifica i dati da inviare eseguendo un OR esclusivo. Questa codifica non è sicura, ma modifica i dati in modo che non possano essere visualizzati. Invece, MQe3DESCryptor implementa DES triplo. Il tipo più semplice di compressore è MQeRleCompressor, che comprime i dati sostituendo i caratteri ripetuti con un conteggio. Vengono forniti anche altri strumenti di autenticazione, compressori e programmi di codifica, vedere Tabella 15 a pagina 49.

Per la scelta dei canali si utilizzano le specifiche dell'adattatore di protocollo per determinare i collegamenti e i protocolli da utilizzare per un determinato canale. Ad ogni nodo intermedio si cercano le definizioni di canale per eseguire l'indirizzo necessario per il collegamento successivo. Quando non ci sono più definizioni, il canale termina e tutti i messaggi in corso vengono spostati al programma di gestione code di quel punto determinato. In questo modo, è possibile utilizzare le code intermedie di archiviazione e le definizioni di coda remota.

I canali non sono direttamente visualizzabili per applicazioni o amministratori e vengono stabiliti dal programma di gestione code come richiesto. I canali collegano tra loro i programmi di gestione code e le relative caratteristiche vengono ripetutamente negoziate da MQSeries Everyplace in base alle

## canali dinamici

informazioni da inviare. I trasportatori sono dei componenti di MQSeries Everyplace che sfruttano i canali per fornire comunicazioni a livello di coda. Anche questi non sono visualizzabili a un programmatore di applicazioni o a un amministratore.

Quando si richiede una messaggistica garantita MQSeries Everyplace consegna i messaggi all'applicazione una, ed una sola, volta. Per fare ciò, assicura che un messaggio venga trasferito correttamente da un programma di gestione code all'altro e che venga confermato, prima di eliminare la copia al termine della trasmissione. Nel caso di un errore di comunicazione, se non c'è stata alcuna conferma, è possibile trasmettere nuovamente un messaggio (la consegna di una volta sola non implica la trasmissione una volta sola) ma non vengono consegnati duplicati.

---

## Adattatori

Gli *adattatori* si utilizzano per creare delle corrispondenze tra MQSeries Everyplace e le interfacce dei dispositivi. I canali sfruttano gli adattatori di protocollo per eseguire HTTP, TCP/IP nativo, UDP ed altri protocolli. Allo stesso modo, le code sfruttano gli adattatori di memoria dei campi come interfaccia per un sistema secondario di memoria, come la memoria o il file system. Adapters forniscono un meccanismo per MQSeries Everyplace ampliare il supporto del dispositivo e per abilitare la conversione.

Un *descrittore di file* è una stringa utilizzata per identificare, caricare e attivare un adattatore.

---

## Gestione dei collegamenti dialup

Il supporto di rete dialup per i dispositivi è gestito dal sistema operativo del dispositivo. Quando MQSeries Everyplace su un dispositivo non collegato tenta di utilizzare la rete, per esempio per inviare un messaggio, se l'accodamento di rete non è attivo, il sistema operativo stesso avvia i RAS (remote access services). In genere viene visualizzato un pannello che propone il profilo per un collegamento dialup. Il sistema operativo esegue un controllo fino a quando viene stabilito il collegamento. Quindi, l'utente del dispositivo deve assicurare che siano disponibili i profili di collegamento dialup adatti al sistema operativo da utilizzare. Non esiste un supporto esplicito per l'Accesso remoto in MQSeries Everyplace Versione 1.2.

---

## Traccia

La traccia viene abilitata avviando un programma indipendente che esegue operazioni di traccia. Sono comprese in MQSeries Everyplace le chiamate per tenere traccia di informazioni, situazioni di errore e avvertimenti con varianti di sistemi e utenti. Le applicazioni possono inoltre richiamare direttamente la

traccia e aggiungere nuovi messaggi o modificare i messaggi di traccia esistenti. Il programma di traccia di esempio fornito permette di visualizzare i messaggi selezionati, stampati e/o diretti alla registrazione eventi. È possibile scrivere altri programmi di traccia con ulteriori funzioni oppure progettarli per formattare e consegnare il risultato in modo diverso.

La maggior parte delle eccezioni di MQSeries Everyplace passa alle applicazioni per la gestione, e il programma di gestione eccezioni può istradarle nella traccia.

---

## Registrazione eventi

MQSeries Everyplace fornisce meccanismi e interfacce di registrazione eventi da utilizzare per lo stato di registrazione, ad esempio per il programma di gestione code avviato. È possibile avviare la registrazione e scriverla in un file, come da impostazione predefinita, ma potrebbe essere intercettata e diretta altrove. La registrazione eventi di MQSeries Everyplace non registra dati di messaggi e non può essere utilizzata per recuperare messaggi o code.

---

## Consegna dei messaggi

Le reti di MQSeries Everyplace sono collegate ai programmi di gestione code e possono includere i gateway. Possono estendersi su più reti fisiche e instradare messaggi tra di esse. In generale, forniscono accesso sincrono e asincrono alle code con un modello di programmazione indipendente dalla posizione della coda.

### Consegna asincrona dei messaggi

Quando un messaggio viene spostato su una coda remota in modo asincrono, l'oggetto messaggio viene spostato logicamente nella memoria di riserva associata alla definizione locale della coda, insieme al programma di gestione code di destinazione e ai nomi di code, e con le funzioni di compressore, strumento di autenticazione e programma di codifica che corrispondono alla destinazione prescelta per il messaggio. Il metodo dump dell'oggetto viene richiamato perché l'oggetto viene salvato nella memoria permanente in un formato protetto, come definito dalla coda di destinazione. Il programma di gestione code controlla la spedizione dei messaggi: identifica (o stabilisce) un canale con le funzioni adatte al programma di gestione code per il passaggio successivo, quindi crea (o riutilizza) un trasportatore per la coda di destinazione. Il trasportatore esegue il dump dell'oggetto e trasmette la stringa di byte di risultato. Il programma di gestione code di destinazione e il nome della coda non fanno parte di quel flusso di messaggio.

Se necessario, il messaggio viene codificato e compresso sul canale. Se raggiunge il programma di gestione code di destinazione, viene decodificato e decompresso. Viene creato quindi un nuovo oggetto messaggio, con il metodo

## consegna dei messaggi

restore di quella classe di oggetto, e il risultato verrà posizionato nella coda di destinazione. Se il messaggio non ha raggiunto il programma di gestione code di destinazione, viene decodificato e decompresso, quindi viene ubicato su una coda di memorizzazione e inoltrato con le caratteristiche appropriate per questo tipo di trasmissione. In entrambi i casi resta sulla coda in un formato protetto, come definito dalla coda di destinazione.

Una caratteristica della consegna sincrona dei messaggi sta nel fatto che i messaggi vengono spostati al programma di gestione con dei passaggi intermedi, essendo accodati per la trasmissione in avanti. I messaggi vengono poi spostati dalle code intermedie prima in ordine di priorità, quindi per sequenza data/ora.

### Consegna sincrona dei messaggi

La consegna sincrona dei messaggi è simile a quella di tipo asincrono descritta sopra, ma il coinvolgimento del programma di gestione code a passaggi intermedi ha luogo ad un livello più inferiore, coinvolgendo anche il trasportatore e i canali. Un canale viene stabilito end-to-end, utilizzando gli adattatori definiti nelle specifiche di protocollo ad ogni nodo intermedio, per identificare il collegamento successivo. Al termine dell'ultimo collegamento, dove non esistono altri descrittori file rilevanti, il messaggio passa ai livelli superiori per essere elaborato dal programma di gestione code. Il nodo di invio quindi non accoda il messaggio ma lo invia lungo il canale tramite passaggi intermedi, per poi passarlo al programma di gestione code prescelto per inviarlo alla coda di destinazione.

Il collegamento a MQSeries utilizza una coda di bridge al gateway, che trasforma il messaggio in un formato MQSeries. Questo meccanismo implica che la messaggistica MQSeries Everyplace di tipo sincrono da un dispositivo è consentita per MQSeries, con il canale dinamico che termina con il gateway. Il messaggio viene recapitato in tempo reale dal gateway, tramite un canale client, ad un server MQSeries. Da qui potrebbe essere necessario istradarlo in modo asincrono su canali di messaggi MQSeries

Allo stesso modo, un dispositivo per sola messaggistica sincrona può inviare messaggi a una coda MQSeries Everyplace asincrona, sempre che sia disponibile un intermediario adatto.

---

## Sicurezza

MQSeries Everyplace fornisce una serie integrata di funzioni di sicurezza che consente di proteggere i dati dei messaggi sia in locale che nel trasferimento.

Le funzioni di sicurezza di MQSeries Everyplace forniscono protezione in tre categorie diverse:

- Sicurezza locale - protezione locale dei dati di messaggio (e altro)

- Sicurezza basata sulle code - protezione dei messaggi tra il programma di gestione code di avvio e la coda di destinazione
- Sicurezza a livello dei messaggi - protezione dei messaggi tra il mittente e il destinatario

La sicurezza locale e a livello di messaggi di MQSeries Everyplace viene utilizzata internamente in MQSeries Everyplace, ma è disponibile anche per altre applicazioni di MQSeries Everyplace. La sicurezza MQSeries Everyplace basata sulle code è un servizio interno.

Le funzioni di sicurezza MQSeries Everyplace di tutte e tre le categorie proteggono i dati dei messaggi utilizzando un attributo (MQAttribute o attributo relativo). A seconda della categoria, l'attributo si applica in modo esplicito o implicito.

Ogni attributo può contenere i seguenti oggetti:

- Strumento di autenticazione
- Programma di codifica
- Compressore
- Chiave
- Nome entità di destinazione

Questi oggetti vengono utilizzati in modo diverso, a seconda della categoria di sicurezza MQSeries, ma in tutti i casi la protezione delle funzioni di sicurezza MQSeries Everyplace è attiva quando viene richiamato l'attributo collegato a un oggetto messaggio. Ciò avviene quando viene richiamato un metodo 'dump' di messaggio di MQSeries Everyplace (quando si utilizza il metodo 'encodeData' dell'attributo, ad esempio per codificare e comprimere i dati di messaggi). Le funzioni di sicurezza di MQSeries Everyplace vengono disabilitate quando si richiama il metodo 'restore' dei messaggi di MQSeries Everyplace (quando si utilizza il metodo 'decodeData' dell'attributo, ad esempio per decomprimere e decodificare i dati di messaggi).

Gli algoritmi supportati da MQSeries Everyplace Versione 1.2 per l'autenticazione, la codifica e la compressione sono descritti in dettaglio in Tabella 15.

*Tabella 15. Supporto per autenticazione, codifica e compressione*

Funzione	Algoritmo
Autenticazione	WTLS mini-certificati
	Convalida identità Windows NT/2000, AIX o Solaris

Tabella 15. Supporto per autenticazione, codifica e compressione (Continua)

Funzione	Algoritmo
Compressione	LZW
	RLE
Codifica	DES triplo
	DES
	MARS
	RC4
	RC6
	XOR

### Sicurezza locale MQSeries Everyplace

La sicurezza locale protegge i dati di messaggi di MQSeries Everyplace (o MQeFields o derivanti da MQeFields) in locale. Si ottiene creando un attributo con programma di codifica simmetrica e compressore adatti, creando e impostando una *chiave* adatta (fornendo una password o una passphrase). La chiave viene esplicitamente collegata all'attributo collegato al messaggio MQSeries Everyplace. MQSeries Everyplace fornisce la classe MQeLocalSecure per l'impostazione della sicurezza locale, ma in tutti i casi, è responsabilità dell'utente della sicurezza locale (interno a MQSeries Everyplace oppure un'applicazione MQSeries Everyplace) impostare un attributo adatto e gestire la chiave per la password oppure per la passphrase.

### Sicurezza MQSeries Everyplace basata sulle code

La sicurezza basata sulle code si applica ai messaggi sincroni e asincroni.

#### Sicurezza basata sulle code sincrone

L'utilizzo della sicurezza basata sulle code sincrone consente a un'applicazione di lasciare tutte le considerazioni sulla sicurezza messaggi a MQSeries Everyplace. Le code dispongono di funzioni di autenticazione, codifica e compressione, che si utilizzano per determinare il livello di sicurezza necessario per proteggere i flussi di messaggi (anche per la memoria permanente).

Quando occorre inviare un messaggio, le funzioni di sicurezza della coda di destinazione vengono richiamate dal registro locale. Se non esistono, il programma di gestione code cerca di scoprire le caratteristiche del programma di destinazione e le richiama per un utilizzo successivo. Se esiste un canale per questo programma di gestione code, verrà utilizzato; altrimenti si creerà un nuovo canale. Vengono recuperati gli attributi delle code di destinazione.

In base alla qualità del servizio richiesto, gli attributi del canale per il programma di gestione code prescelto vengono modificati in modo dinamico.

Il programma è soggetto a tutte le regole stabilite. In genere una regola consente un aggiornamento al livello di sicurezza, per esempio dal livello senza protezione a quello di bassa protezione, o da una protezione bassa ad una elevata. Se non è possibile aggiornare il canale, o il livello di sicurezza è giudicato eccessivo (per esempio non è richiesta alcuna protezione e il canale disponibile implementa una protezione forte) verrà creato un nuovo canale. Esiste una serie di canali, riutilizzati quando è possibile, con caratteristiche che si modificano dinamicamente, in base alle domande del traffico. I canali vengono distrutti automaticamente quando non sono richiesti. I messaggi vengono sempre posizionati sulle code al livello di sicurezza definito dalle caratteristiche della coda di destinazione.

L'autenticazione avviene a livello del canale, tenendo il livello minimo per messaggio. La sicurezza basata sulle code sincrone viene in genere utilizzata anche con i programmi di codifica simmetrica poiché consente una codifica/decodifica rapida. Nei casi simmetrici, MQSeries Everyplace utilizza all'inizio la codifica asimmetrica RSA per proteggere i flussi necessari per stabilire una chiave condivisa per il mittente e il destinatario. In seguito la codifica simmetrica viene utilizzata per proteggere la segretezza dei dati trasmessi. MQSeries Everyplace rende più impenetrabile la codifica di questi dati in quanto modifica in modo dinamico la chiave a ciascun flusso del canale. MQSeries Everyplace assicura inoltre l'integrità dei dati trasmessi creando e collegando un riassunto ai dati prima dell'invio e convalidandolo alla ricezione.

### **Sicurezza basata sulle code asincrone**

La messaggistica asincrona differisce da quella sincrona sopra descritta in quanto non fornisce una garanzia che la coda di destinazione sia accessibile nel momento in cui viene eseguita la funzione `putMessage`. In tal caso il programma di gestione code non potrà inviare immediatamente il messaggio e lo posizionerà sulla coda di trasmissione; tuttavia verrà codificato in base alle caratteristiche della coda prescelta. Nel momento in cui è possibile trasmetterlo, viene decodificato e poi inviato ad un canale con le caratteristiche adatte. I messaggi quindi sono sempre protetti, anche in attesa della trasmissione. La messaggistica asincrona richiede la definizione della coda remota per determinare le caratteristiche della coda di destinazione.

Per il tipo asincrono, l'autenticazione non è possibile tra l'elemento di origine e di destinazione. Laddove l'autenticazione è importante, per esempio quando il destinatario deve determinare il mittente del messaggio (per stabilire se accettare o meno) o quando il mittente deve assicurare che il messaggio giunga solo al destinatario prescelto, è necessario utilizzare la sicurezza a livello di messaggio.

## sicurezza

La sicurezza basata sulle code si utilizza insieme alla sicurezza a livello di messaggio, ma non è necessaria in quanto i dati di messaggio sono già protetti.

### **Sicurezza a livello di messaggio**

La sicurezza a livello di messaggio assicura la protezione dei dati di messaggio tra l'applicazione MQSeries Everyplace del mittente e quella del destinatario.

La sicurezza a livello di messaggio è un servizio per applicazioni che richiede che l'applicazione iniziale MQSeries Everyplace imposti un attributo a livello di messaggio e lo fornisca quando si utilizza la funzione `putMessage` per posizionare il messaggio su una coda di destinazione. L'applicazione per la ricezione deve impostare e poi passare un attributo corrispondente a livello di messaggio al programma di gestione code che lo riceve, in modo che l'attributo sia disponibile quando l'applicazione richiama `getMessage` per ricevere il messaggio dalla coda di destinazione.

Come la sicurezza locale, la sicurezza a livello di messaggio sfrutta l'applicazione dell'attributo in un oggetto messaggio. Il programma di gestione code dell'applicazione di invio gestisce la funzione `putMessage` con il metodo `'dump'`, che utilizza il metodo dell'attributo `'encodeData'` per proteggere i dati dei messaggi. Il programma di gestione code dell'applicazione per la ricezione gestisce la funzione dell'applicazione `getMessage` con il metodo `'restore'` che utilizza a sua volta il metodo dell'attributo `'decodeData'` per recuperare i dati del messaggio originale.

MQSeries Everyplace fornisce due attributi alternativi per la sicurezza a livello di messaggio:

#### **MQeMAttribute**

Si utilizza per comunicazioni tra aziende in cui la fiducia reciproca viene gestita a livello delle applicazioni e non è richiesta una terza parte in qualità di garante della sicurezza. È possibile utilizzare tutte le opzioni disponibili per il programma di codifica simmetrica e il compressore di MQSeries Everyplace. Come per la sicurezza locale, la chiave dell'attributo deve essere preimpostata prima di fornirla per le funzioni `putMessage` o `getMessage`. MQeAttribute fornisce un metodo semplice ed efficace per la protezione a livello di messaggi, consentendo l'utilizzo della codifica per proteggere la riservatezza dei messaggi, senza alcuna PKI (public key infrastructure) superiore.

#### **MQeMTrustAttribute**

Questo attributo fornisce una soluzione molto più avanzata, che utilizza firme digitali e sfrutta l'infrastruttura della chiave pubblica predefinita. Utilizza infatti la firma digitale e la convalida ISO9796 per abilitare l'applicazione per la ricezione a provare che il messaggio

proviene effettivamente dal mittente dichiarato. Il programma di codifica dell'attributo fornito si utilizza per proteggere la riservatezza del messaggio. Il riassunto SHA1 garantisce l'integrità dei messaggi e la codifica/decodifica RSA assicura che il messaggio può essere recuperato solo dal destinatario. Come per MQeAttribute, è possibile utilizzare tutte le opzioni disponibili per il programma di codifica simmetrica e il compressore di MQSeries Everyplace. Scelti per un'ottimizzazione delle dimensioni, i certificati utilizzati sono di tipo WTLS mini-certificati. La struttura predefinita di MQSeries Everyplace fornisce la reciproca disponibilità delle informazioni necessarie per autenticare (convalidare firme) e codificare/decodificare.

Un messaggio tipico protetto da MQeMTrustAttribute avrà il formato:

```
RSA-enc{SymKey}, SymKey-enc {Data, DataDigest,
DataSignature}
```

dove:

<b>RSA-enc:</b>	RSA codificata con la chiave pubblica del destinatario dichiarato
<b>SymKey</b>	chiave simmetrica creata in modo casuale
<b>SymKey-enc</b>	codificata simmetricamente con SymKey
<b>Data</b>	dati del messaggio
<b>DataDigest</b>	riassunto dati del messaggio
<b>DigSignature</b>	firma digitale del mittente del messaggio

La sicurezza a livello di messaggio è indipendente dalla sicurezza a livello di coda.

## Il registro

Il registro è il primo magazzino di informazioni relative al programma di gestione code; ne esiste uno per ogni programma. Ogni programma utilizza il registro per memorizzare:

- Dati di configurazione dei programmi di gestione code
- Definizioni di code
- Definizioni di code remote
- Definizioni di programmi di gestione code remote
- Dati utente (comprese informazioni di sicurezza in base alla configurazione)

L'accesso al registro è in genere limitato all'utente legittimato del relativo programma e il PIN è protetto, ma con un'opzione configurabile è possibile dare l'accesso ad utenti specializzati in dimensioni piuttosto che in sicurezza.

### Entità autenticabili MQSeries Everyplace

La sicurezza basata sulle code, che utilizza una reciproca autenticazione con mini-certificati e la protezione a livello di messaggi, che utilizza la forma digitale, hanno introdotto il concetto di *entità autenticabile*. Nel caso dell'autenticazione reciproca, è normale pensare all'autenticazione tra due utenti (persone), ma in generale la messaggistica non si riferisce a un utente. Il concetto è invece gestito a livello delle applicazioni, quindi dall'utente di servizi di messaggistica. MQSeries Everyplace astrae deliberatamente il concetto di *destinazione di autenticazione* dall'utente per *entità autenticabile*. Non si esclude la possibilità che le entità autenticabili siano delle persone, tuttavia sarebbe una corrispondenza tra applicazioni. All'interno, MQSeries Everyplace definisce tutti i programmi di gestione code che possono creare o essere la destinazione di servizi dipendenti da mini-certificati come entità autenticabili. Inoltre, MQSeries Everyplace definisce le code stabilite per utilizzare gli strumenti di autenticazione basati su mini-certificati come entità autenticabili. Quindi, i programmi di gestione code che supportano questi servizi, possono avere un'entità autenticabile, un programma di gestione di code, oppure una serie di entità autenticabili, un programma di gestione di code e tutte le code che utilizzano un programma di autenticazione basato su certificato.

### Registro privato e credenziali

Ogni entità autenticabile, per poter essere utilizzata, ha bisogno delle proprie credenziali. Ciò implica due obiettivi. Prima di tutto, come seguire la registrazione per ottenere le credenziali, e poi dove gestire le credenziali in modo sicuro. In genere questi obiettivi sono più difficili da attuare rispetto alle tecniche di codifica. MQSeries Everyplace fornisce servizi predefiniti che possono essere utilizzati per consentire alle entità autenticabili di eseguire l'auto registrazione. Consente inoltre, al registro privato, ( un elemento secondario del registro di base) di gestire in modo sicuro le credenziali private di un'entità autenticabile, e al registro pubblico (altro elemento secondario del registro di base) di gestire le credenziali pubbliche. Il registro privato fornisce al registro di base vari tipi di simboli sicuri o codificati, ad esempio, può essere un luogo sicuro per conservare oggetti pubblici come mini-certificati e oggetti privati come chiavi private. Fornisce inoltre un meccanismo che consente unicamente all'utente autorizzato di accedere agli oggetti privati e un supporto per servizi come la firma digitale e la codifica RSA, per cui gli oggetti privati non possano mai essere spostati dal registro privato. Grazie a un'interfaccia comune, nasconde il supporto del dispositivo, che è attualmente limitato al sistema di file locale, ma può essere esteso per delle corrispondenze future.

### Autoregistrazione

MQSeries Everyplace fornisce servizi predefiniti che supportano l'autoregistrazione. Questi servizi vengono avviati automaticamente quando si configura un'entità autenticabile, per esempio quando si avvia un programma di gestione code o si definisce una nuova coda. In entrambi i casi viene

avviata la registrazione, vengono create nuove credenziali e memorizzate nel registro privato dell'entità autenticabile. Le fasi di autoregistrazione includono la creazione di una nuova coppia di chiavi RSA, la protezione e il salvataggio della chiave privata nel registro privato e la sistemazione della chiave pubblica in una richiesta di *nuovo certificato* al server dei mini-certificati predefinito. Supponendo che server dei mini-certificati sia configurato e disponibile, restituisce il nuovo mini-certificati dell'entità autenticabile insieme al proprio mini-certificati e questi, insieme alla chiave privata protetta, vengono memorizzate nel registro privato dell'entità autenticabile come nuove credenziali. Mentre l'autoregistrazione fornisce un meccanismo semplice per stabilire le credenziali dell'entità autenticabile, per la protezione a livello di messaggio (MQeMTrustAttribute, vedere sopra), è necessario anche l'accesso alla chiave pubblica del destinatario (mini-certificati).

### **Registro pubblico e replica di certificati**

MQSeries Everyplace fornisce dei servizi predefiniti per la condivisione delle credenziali pubbliche di entità autenticabili (mini-certificati) tra i componenti Everyplace di MQSeries. Questi costituiscono un requisito essenziale per la sicurezza a livello di messaggio basata su MQeMTrust. Il registro pubblico di MQSeries Everyplace fornisce un magazzino accessibile a tutti per i mini-certificati. Costituisce una funzione analoga a quella della rubrica telefonica personale su un telefono cellulare: la differenza sta nel fatto che, invece dei numeri di telefono, si tratta di una serie di mini-certificati delle entità autenticabili che vengono contattate più frequentemente. Il registro pubblico non è del tutto passivo per i servizi. Se si accede per fornire un mini-certificati che non è presente, e il registro è configurato con un componente del server home-server valido, il registro pubblico tenta automaticamente di recuperare il mini-certificati richiesto dal registro pubblico del server home-server. È possibile utilizzare questi servizi per fornire un servizio di replica automatizzata intelligente di mini-certificati che consente di avere a disposizione il mini-certificati giusto al momento giusto.

### **Utilizzo applicazioni dei servizi di registro**

Il programma di gestione code di MQSeries Everyplace è progettato per utilizzare al meglio i servizi di registro privato e pubblico, e l'accesso a questi servizi non è limitato. Le soluzioni MQSeries Everyplace potrebbero definire e gestire le proprie entità autenticabili, per esempio gli utenti. È possibile inoltre utilizzare i servizi di registro privato per autoregistrare e gestire le credenziali delle nuove entità autenticabili, e i servizi di registro pubblico per rendere disponibili in qualsiasi momento le credenziali pubbliche. Tutte le entità autenticabili registrate possono essere utilizzate come servizi a livello di messaggio del mittente o del destinatario, protetti da MQeMTrustAttribute

### **Servizi di creazione predefiniti di mini-certificati**

MQSeries Everyplace fornisce un servizio di creazione mini-certificati predefinito da configurare per poter soddisfare richieste di autoregistrazione registro privato. Con gli strumenti forniti da MQSeries Everyplace, una

soluzione è in grado di impostare e gestire un servizio per creare mini-certificati per una serie ben controllata di nomi di entità. Le caratteristiche di questo servizio sono:

- Gestione dell'insieme di entità autenticabili registrate
- creazione di mini-certificati
- Gestione magazzino dei mini-certificati

Gli strumenti forniti con MQSeries Everyplace permettono all'amministratore dei servizi di creazione di mini-certificati di autorizzare la creazione di un mini-certificati per una determinata entità, registrando il nome dell'entità e l'indirizzo, e definendo un PIN di richiesta certificati valido per una sola operazione. L'operazione viene eseguita dopo che il controllo 'offline' abbia convalidato l'autenticità del richiedente. Il PIN di richiesta certificati viene inviato all'utente definito (in modo simile all'invio del PIN di una nuova carta di credito). È possibile quindi configurare l'utente del registro privato (ad esempio il programma di gestione applicazioni MQSeries Everyplace o code MQSeries Everyplace) per fornire questo PIN di richiesta certificati al momento dell'avvio. Quando il registro privato avvia l'autoregistrazione, il servizio di creazione di mini-certificati convalida le nuove richieste di certificati (basate sulla corrispondenza tra il nome dell'entità e il PIN di richiesta certificati con i corrispondenti valori preregistrati), crea il nuovo mini-certificati e reimposta il PIN di richiesta certificati in maniera tale da non poter essere riutilizzato. Tutte le richieste di autoregistrazione di nuovi mini-certificati vengono eseguite su un canale sicuro.

La serie di mini-certificati emessi da un apposito servizio di creazione viene conservata nel relativo registro. Quando un mini-certificati viene riemesso (ad esempio dopo una scadenza), mini-certificati scaduto viene archiviato.

### L'interfaccia di sicurezza

Viene fornita un'interfaccia facoltativa da implementare in un programma di gestione sicurezza personalizzata. I metodi consentono al programma di gestione sicurezza di autorizzare o respingere le richieste associate a:

- Aggiunta o rimozione di alias di classi
- Definizione di adattatori
- Corrispondenza di descrittori file
- Esecuzione di comandi di canali

---

## Personalizzazione

### Regole

Le regole sono classi Java utilizzate per personalizzare l'utilizzo di MQSeries Everyplace in caso di modifiche di stato. Vengono fornite delle regole predefinite laddove necessario, ma possono essere sostituite da regole di

applicazioni, o specifiche per installazione, per soddisfare i requisiti del cliente. I tipi di regole supportate differiscono nel modo in cui vengono avviate, ma non nelle prestazioni. Le regole contengono una logica per cui sono in grado di svolgere un'ampia gamma di funzioni.

### **Regole di attributo**

Questa classe di regole viene controllata ogni volta che si tenta di apportare una modifica di stato, per esempio a:

- Strumento di autenticazione
- Compressore
- Programma di codifica

La regola in genere abilita o meno la modifica.

### **Regole di bridge MQSeries**

Queste classi di regole vengono controllate quando MQSeries Everyplace per il codice di collegamento MQSeries-bridge modifica lo stato. Esiste una classe separata per le regole di bridge per determinare quanto segue:

- Cosa fare di un messaggio quando non è possibile spostarlo in MQSeries Everyplace, se proviene da MQSeries. Ad esempio perché il messaggio è troppo grande o la coda non esiste.
- Lo stato in cui gli oggetti amministrati da bridge dovrebbero partire quando viene creata l'istanza del server
- Cosa fare quando il bridge rileva un problema con la coda Sync su MQSeries (la memoria permanente utilizzata per il recupero errori). La regola predefinita visualizza solo il problema.
- Come convertire un messaggio MQSeries Everyplace in un messaggio MQSeries e viceversa. I trasformatori per eseguire la conversione di messaggi tra MQSeries Everyplace e MQSeries non derivano da classi MQeRule, ma devono implementare l'interfaccia MQeTransformerInterface. Inoltre, i trasformatori funzionano come le regole e vengono richiamati quando un messaggio richiede la conversione del formato.

### **Regole di coda**

Questa classe di regole viene controllata ogni volta che si verifica una modifica di stato della coda associata, per esempio:

- Aggiunta di un messaggio su una coda. Per esempio, per visualizzare se la soglia di errore viene superata (numero di messaggi, dimensione del messaggio, priorità non valida)
- Caratteristiche della coda assegnata o modificata
- La coda viene aperta o chiusa
- La coda deve essere eliminata

## personalizzazione

### Regole del programma di gestione code

Questa classe di regole viene controllata ogni volta che si verifica una modifica di stato del programma di gestione code, per esempio:

- Il programma di gestione code viene aperto. Per esempio, avviare un thread del timer per abilitare operazioni a tempo
- Il programma di gestione code viene chiuso. Per esempio, chiudere il thread del timer
- Viene aggiunta una nuova coda

---

## Stili di collegamento

MQSeries Everyplace può supportare client-server e operazioni peer-to-peer. Un *client* è in grado di avviare una comunicazione con un server. Un *server* è solo in grado di rispondere alle richieste da parte di un client. In un'operazione *peer-to-peer*, i due pari (peer) possono avviare flussi in entrambe le direzioni. Questi stili di collegamento hanno bisogno di diversi componenti di MQSeries Everyplace attivi. I componenti coinvolti sono:

- **Listener di canali:** in attesa di richieste di collegamento in entrata.
- **Programma di gestione canali:** che supporta più cavi di comunicazione logica in contemporanea tra le estremità.
- **Gestore code:** che supporta le applicazioni grazie a funzioni di messaggistica e di accodamento.

Tabella 16 mostra la relazione tra questi componenti e lo stile di collegamento. Lo stile di collegamento client-server descrive la situazione in cui MQSeries Everyplace può funzionare in modo client o server. L'opzione servlet descrive il caso in cui MQSeries Everyplace è configurato come servlet HTTP con il server stesso responsabile per le richieste di collegamento in entrata.

Tabella 16. Stili di collegamento

	Gestore code	Programma di gestione canali	Listener di canali
Client	Sì		
Client-server	Sì	Sì	Sì
Pari	Sì		
Server	Sì	Sì	Sì
Servlet	Sì	Sì	

Le applicazioni MQSeries Everyplace non sono direttamente responsabili dello stile di collegamento adottato dai gestori code. Lo stile è però importante perché influenza le risorse disponibili per le parti, che i programmi di gestione code possono collegare ad altri programmi, la struttura di MQSeries Everyplace e i collegamenti che possono esistere contemporaneamente.

## Collegamento peer-to-peer

Un canale peer-to-peer comprende le funzioni di un programma di gestione canali e un listener per un singolo canale. Quando si crea un canale peer-to-peer tra due programmi di gestione code, uno di questi deve funzionare come listener e l'altro come punto di avvio del collegamento. Un programma di gestione code collegato peer-to-peer può avviare più collegamenti peer-to-peer con altri programmi di gestione code, ma può rispondere solo a una richiesta di collegamento in entrata e deve attendere la chiusura del canale peer-to-peer prima di poter rispondere a un'altra richiesta simile. Oltre un canale peer-to-peer, i due programmi di gestione code attivi possono avviare delle operazioni, quindi per esempio le applicazioni di ciascun programma possono avere accesso alle code di un altro programma.

I canali peer-to-peer non sono utilizzabili con i firewall perché la destinazione della richiesta di collegamento in entrata non viene accettata.

## Collegamento Client-server

I canali standard, utilizzati per lo stile di collegamento client-server, non hanno funzioni di attesa ma dipendono da un listener indipendente al server, e il server richiede un programma di gestione canale per gestire più canali contemporanei. Il client avvia la richiesta di collegamento e il server risponde. Un canale server può in genere gestire più richieste in entrata dai client. Oltre ad un canale standard, il client ha accesso alle risorse sul server. Se un'applicazione sul server ha bisogno dell'accesso sincrono alle risorse sul client, è necessario un secondo canale quando le regole vengono invertite. Tuttavia, essendo i canali standard di tipo bidirezionale, i messaggi destinati a un client dalla coda di trasmissione del server, vengono trasmessi sul canale standard (client-server) avviato.

Un client deve essere abilitato come tale per più server contemporaneamente. Non è richiesto un programma di gestione canali per supportare questa configurazione poiché questi programmi gestiscono più canali in entrata.

Lo stile di collegamento client-server è in genere adatto per l'utilizzo con i firewall, in quanto la destinazione della richiesta di collegamento in entrata viene normalmente accettata.

## Stili di collegamento multipli

Un programma di gestione code singolo può avviare collegamenti peer-to-peer oppure client-server, e può rispondere a un server o a un pari (peer). In questo caso, il listener di canali di pari e il listener di canali standard devono avere numeri di porta diversi.

---

### Classi

MQSeries Everyplace fornisce una scelta di classi per determinate funzioni, per poter personalizzare MQSeries Everyplace aderendo ai requisiti specifici delle applicazioni. In alcuni casi le interfacce delle classi sono documentate per poter sviluppare ulteriori alternative. Tabella 17 a pagina 61 riassume le possibilità. Le classi possono essere identificate in modo esplicito oppure utilizzando degli alias.

Tabella 17. Opzioni di classe

Classe	Alternative fornite	Interfacce documentate
amministrazione	no	sì
Strumenti di autenticazione	sì	no
Adattatore di comunicazioni	sì	sì
Stili di comunicazione	sì	no
Compressori	sì	no
Programmi di codifica	sì	no
Registrazione eventi	campione fornito	sì
Messaggi	no	sì
Memoria delle code	sì	no
Regole	classi predefinite fornite	sì
Traccia	campioni forniti	sì

## Caricamento applicazioni

Quando MQSeries Everyplace è configurato per funzionare come client (o peer), l'applicazione di avvio è responsabile del caricamento di tutte le altre applicazioni in JVM. Le funzioni Java standard possono essere utilizzate a questo scopo, altrimenti è disponibile lo strumento di caricamento incluso come parte di MQSeries Everyplace. Quindi è possibile eseguire più applicazioni con un singolo programma di gestione code nello stesso JVM. Altrimenti è possibile utilizzare più JVM ma ciascuno richiede il proprio programma di gestione code, ciascuno dei quali deve avere un nome unico.

Quando MQSeries Everyplace è configurato come server, funziona anche come applicazione di avvio. MQSeries Everyplace supporta un elenco prestabilito di classi, che vengono caricate a turno, prima di caricare il programma stesso di gestione code.



---

## Capitolo 6. Reti MQSeries Everyplace e MQSeries

Sebbene possa esistere una rete MQSeries Everyplace indipendente, senza la necessità di un server o di una rete MQSeries, in pratica MQSeries Everyplace viene spesso utilizzato per completare un'installazione esistente di MQSeries, ampliando il raggio a nuove piattaforme e dispositivi, oppure fornendo funzioni avanzate come la sicurezza basata su code o messaggi, o la messaggistica sincrona. Da una prospettiva di applicazione di MQSeries Everyplace, le code e i relativi programmi di gestione di MQSeries possono essere considerati semplicemente come code remote o programmi di gestione code aggiuntivi. Tuttavia, esistono alcune limitazioni funzionali perché non è possibile accedere a queste code direttamente tramite canali dinamici di MQSeries Everyplace e un programma di gestione code di MQSeries Everyplace, ma è necessario un gateway MQSeries Everyplace. Il gateway può inviare messaggi a più programmi di gestione code di MQSeries in modo diretto o indiretto, attraverso canali client di MQSeries. In caso di collegamento indiretto, i messaggi passano attraverso canali client di MQSeries verso un programma di gestione code MQSeries intermedio, quindi prosegue attraverso canali di messaggi MQSeries verso il programma di destinazione.

I messaggi da un'applicazione di MQSeries destinata a MQSeries Everyplace vengono indirizzati al programma di gestione code e alla coda di MQSeries Everyplace in modo normale, con l'instradamento definito di MQSeries (definizioni di programma di gestione code remote), in modo che i messaggi di MQSeries Everyplace arrivino su code di trasmissione specifiche di MQSeries. I canali di MQSeries non sono definiti per le code di trasmissione, come vorrebbe la prassi normale; il gateway di MQSeries Everyplace infatti estrae i messaggi da queste code e ne garantisce la consegna alla destinazione MQSeries Everyplace. Il numero di code di trasmissione da utilizzare (numero di percorsi) è configurabile e dovrebbe essere impostato per riflettere il volume di messaggi da recapitare.

---

### Interfaccia per MQSeries

La struttura di MQSeries Everyplace supporta il concetto di uno o più bridge facoltativi tra MQSeries Everyplace e altri sistemi di messaggistica.

In MQSeries Everyplace Versione 1.2 è supportato un solo tipo di bridge, il *bridge MQSeries* che funziona da interfaccia tra le reti MQSeries Everyplace e MQSeries. Questo bridge utilizza il client Java MQSeries come interfaccia per uno o più gestore code di MQSeries, consentendo ai messaggi di passare da MQSeries Everyplace a MQSeries e viceversa. Nella versione corrente di

## collegamento a MQSeries

MQSeries Everyplace si consiglia un solo collegamento del genere per il server, ciascuno associato a più *proxy di MQSeries* (definizioni di MQSeries). È richiesta una definizione di proxy di programma di gestione code per ciascun programma MQSeries che comunica con MQSeries Everyplace. Ciascuna definizione può avere uno o più *servizi di collegamento client* associati, di cui ognuno rappresenta un collegamento a un singolo programma di gestione code di MQSeries. Ciascuno di questi può utilizzare un diverso collegamento del server MQSeries al programma di gestione code, e inoltre una serie diversa di proprietà come le uscite o porte utenti.

Un gateway può avere un numero di *listener* che utilizzano questo gateway per il collegamento a MQSeries e per il recupero di messaggi da MQSeries a MQSeries Everyplace. Un listener utilizza un solo servizio per stabilire il collegamento, e ciascun listener si collega ad una sola coda di trasmissione sul relativo programma MQSeries. Ogni listener sposta i messaggi da una singola coda di trasmissione di MQSeries a qualsiasi altro punto della rete MQSeries Everyplace, tramite il programma principale di gestione code del gateway. In questo modo, un singolo gestore code del gateway può incanalare i messaggi MQSeries nella rete MQSeries Everyplace.

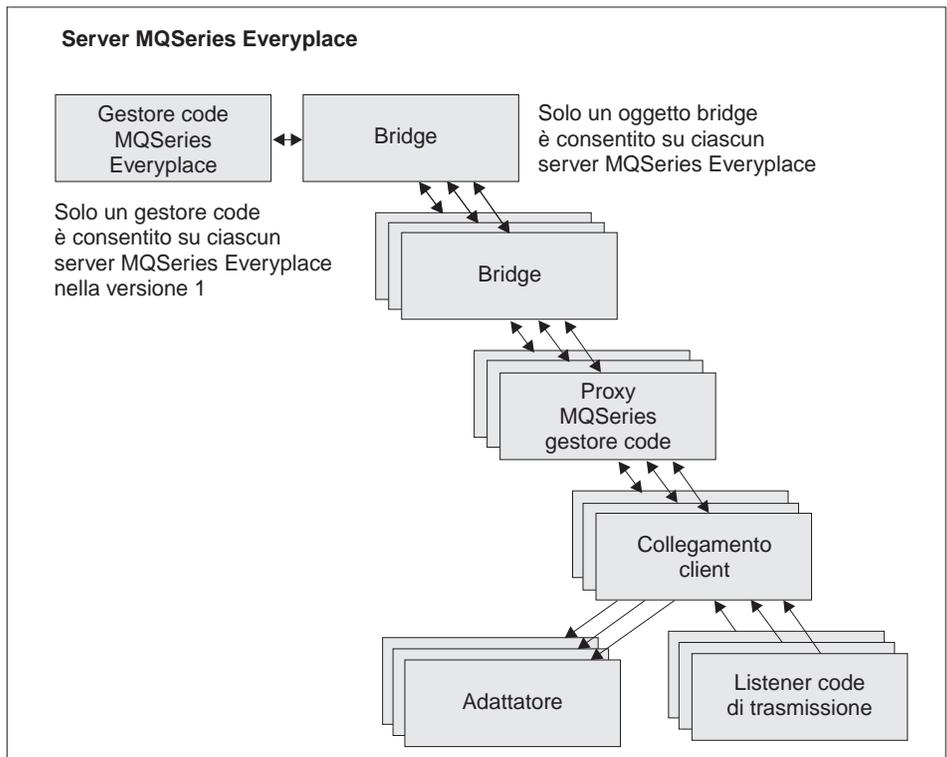


Figura 9. Gerarchia oggetti di bridge MQSeries

Nello spostamento dei messaggi in un'altra direzione, da MQSeries Everyplace a MQSeries, il gestore code del gateway configura uno o più oggetti *code di bridge*. Ciascun oggetto coda di bridge è in grado di collegarsi direttamente a un programma di gestione code e di inviare messaggi alla coda di destinazione. In questo modo, un gateway può distribuire messaggi di MQSeries Everyplace instradati attraverso un singolo programma di code MQSeries Everyplace ad un qualsiasi altro programma simile, in modo diretto o indiretto. L'oggetto bridge contiene le proprietà indicate in Tabella 18.

Tabella 18. Proprietà degli oggetti di bridge

Proprietà	Spiegazione
Nomecollegamento	Elenco nomi collegamenti
Stato esecuzione	Stato: in esecuzione o arrestato

L'oggetto collegamenti e altri oggetti del gateway possono essere avviati e arrestati indipendentemente dal programma di gestione code di MQSeries Everyplace. Se viene avviato o arrestato un oggetto gateway, l'operazione fornisce anche tutti gli elementi secondari (collegamenti, proxy del

## collegamento a MQSeries

programma di gestione, collegamenti al client e listener della code di trasmissione). L'oggetto collegamento contiene le proprietà indicate in Tabella 19.

Tabella 19. Proprietà collegamenti

Proprietà	Spiegazione
Classe	Classe collegamenti
Trasformatore predefinito	La classe predefinita (classe di regole) da utilizzare per trasformare un messaggio da MQSeries Everyplace a MQSeries (o viceversa) se nessun'altra classe di trasformatore è associata alla coda di destinazione
Intervallo impulsi	L'unità di tempo base da utilizzare per operazioni con oggetti di collegamento
Nome	Nome dell'oggetto di collegamento
Stato esecuzione	Stato: in esecuzione o arrestato
Classe regole di avvio	Classe di regole utilizzate quando si avvia un oggetto di collegamento
Elementi secondari proxy del programma di gestione code di MQSeries	Elenco di tutti i proxy dei programmi di gestione code in questo collegamento

In alcuni casi è possibile utilizzare un trasformatore predefinito (regola) per gestire tutte le conversioni dei messaggi. È inoltre possibile impostare un trasformatore su una base listener (per messaggi da MQSeries a MQSeries Everyplace) che ricopre quello predefinito. Per un controllo più specifico, è possibile impostare le regole di trasformazione sulla base di una coda di destinazione utilizzando definizioni di code di collegamento sul gateway. Ciò è valido per le code di destinazione di MQSeries Everyplace e MQSeries.

Il proxy del programma di gestione code MQSeries è dotato delle proprietà specifiche per un singolo programma di gestione code di MQSeries. Le proprietà del proxy sono indicate in Tabella 20.

Tabella 20. Proprietà proxy del programma di gestione code MQSeries

Proprietà	Spiegazione
Classe	classe proxy del programma di gestione code di MQSeries
nome host MQSeries	Il nome host IP utilizzato per stabilire collegamenti al programma di code di MQSeries tramite le classi client Java. Se non viene specificato, si assume che il programma di gestione code di MQSeries si trovi sullo stesso computer del collegamento e dei bind Java

Tabella 20. Proprietà proxy del programma di gestione code MQSeries (Continua)

Proprietà	Spiegazione
nome proxy del programma di gestione code di MQSeries	Il nome del programma di gestione code di MQSeries
Nome del collegamento in corso	Nome dell'oggetto di collegamento che gestisce questo proxy MQSeries
Stato esecuzione	Stato: in esecuzione o arrestato
Classe regole di avvio	Classe di regole utilizzate quando si avvia un oggetto programma di gestione code di MQSeries
Elementi secondari di collegamento al client	Elenco di tutti gli oggetti di collegamento al client gestiti da questo proxy

Il servizio di collegamento al gateway contiene le informazioni dettagliate necessarie per stabilire un collegamento ad un programma di gestione code di MQSeries. Le proprietà del collegamento sono indicate in Tabella 21.

Tabella 21. Proprietà del servizio di collegamento al client

Proprietà	Spiegazione
Classe di adattatori	Classe da utilizzare come adattatore gateway
CCSID*	Il valore CCSID MQSeries da utilizzare
Classe	Classe di servizio di collegamento al client
Tempo massimo di collegamento inattivo	Il tempo massimo in cui un collegamento può restare inattivo prima di essere chiuso
password di MQSeries*	Password da utilizzare per il client Java
porta di MQSeries*	Il numero di porta IP utilizzato per stabilire collegamenti al programma di gestione code di MQSeries tramite le classi client Java. Se non viene specificato, si assume che il programma di gestione code di MQSeries si trovi sullo stesso computer del collegamento e dei bind Java
classe uscita di ricezione MQSeries *	Utilizzata per associare l'uscita di ricezione con l'altra estremità del canale client; l'uscita ha una stringa associata per consentire il passaggio dei dati al codice di uscita
classe uscita di sicurezza MQSeries *	Utilizzata per associare l'uscita di sicurezza con l'altra estremità del canale client; l'uscita ha una stringa associata per consentire il passaggio dei dati al codice di uscita
classe uscita di invio MQSeries *	Utilizzata per associare l'uscita di invio con l'altra estremità del canale client; l'uscita ha una stringa associata per consentire il passaggio dei dati al codice di uscita
MQSeriesID utente*	ID utente per il client Java

## collegamento a MQSeries

Tabella 21. Proprietà del servizio di collegamento al client (Continua)

Proprietà	Spiegazione
Nome servizio di collegamento al client	Nome del canale di collegamento al server per il computer MQSeries
Nome del proxy del programma di gestione code corrente	Il nome proxy del programma di gestione code corrente
Classe regole di avvio	Classe di regole utilizzate quando si avvia un oggetto servizio di collegamento al client
Nome coda Sync	Il nome della coda di MQSeries utilizzata dal collegamento per scopi di sincronizzazione
Classe di regole per coda di sincr.	La classe di regole da utilizzare quando un messaggio si trova su una coda di sincr.
Stato esecuzione	Stato: in esecuzione o arrestato
Nome del collegamento in corso	Il nome dell'oggetto di collegamento che gestisce questo collegamento al client
Elementi secondari del listener MQ XmitQ	Elenco di tutti gli oggetti del listener che utilizzano questo collegamento al client
*I dettagli su questi parametri sono reperibili nella documentazione <i>MQSeries Using Java</i>	

La *classe di adattatore* viene utilizzata per inviare messaggi da MQSeries Everyplace a MQSeries, mentre la *coda di sincr.* serve per tenere traccia dello stato dell'operazione. Il contenuto si utilizza in situazioni di recupero della messaggistica assicurata; dopo una normale chiusura la coda risulterà vuota. Può essere condivisa tra più collegamenti al client e tra più definizioni di collegamenti, purché le uscite di ricezione, invio e sicurezza siano le stesse. Questa coda si utilizza anche per memorizzare lo stato dei messaggi che vengono spostati da MQSeries a MQSeries Everyplace, a seconda delle proprietà del listener utilizzato. La *classe di regole per coda di sincr.* si utilizza quando un messaggio si trova sulla coda di sincr. e indica un errore di MQSeries Everyplace per confermare un messaggio.

Il tempo massimo di collegamento inattivo viene utilizzato per controllare l'insieme di collegamenti al client Java gestiti dal servizio di collegamenti al client per il sistema MQSeries. Quando un collegamento a MQSeries diventa inattivo, per mancanza di utilizzo, viene avviato un timer: il collegamento inattivo si chiuderà se il timer scatta prima che il collegamento venga riutilizzato. La creazione di collegamenti a MQSeries è un'operazione costosa, quindi questa procedura assicura che i collegamenti vengano riutilizzati senza consumare troppe risorse. Un valore zero indica che l'insieme di collegamenti non dovrebbe essere utilizzato.

L'oggetto listener, che passa i messaggi da MQSeries a MQSeries Everyplace, è dotato delle proprietà indicate in Tabella 22.

Tabella 22. Proprietà listener

Proprietà	Spiegazione
Classe	Classe listener
Nome coda delle lettere scadute	Coda utilizzata per conservare i messaggi da MQSeries a MQSeries Everyplace che non possono essere recapitati
Adattatore di memoria stato del listener	Nome classe dell'adattatore utilizzato per memorizzare informazioni sullo stato
Nome listener	Nome dei messaggi nella coda XMIT di MQSeries
Nome servizio corrente di collegamento al client	Nome servizio di collegamento al client
Stato esecuzione	Stato: in esecuzione o arrestato
Classe regole di avvio	Classe di regole utilizzate quando si avvia un oggetto listener
Classe trasformatore	Classe di regole utilizzate per determinare la conversione di un messaggio MQSeries ad un messaggio MQSeries Everyplace
Classe regole di messaggi non recapitati	Classe di regole utilizzata per stabilire cosa fare quando non è possibile recapitare messaggi da MQSeries a MQSeries Everyplace
Secondi di attesa per il messaggio	Un'opzione avanzata per controllare le prestazioni del listener in casi particolari

La *classe di regole messaggi non recapitati* determina cosa fare quando non è possibile recapitare un messaggio da MQSeries a MQSeries Everyplace. In genere viene posizionata nella *coda lettere scadute* nel sistema MQSeries.

Per garantire la consegna dei messaggi, la classe del listener utilizza un *adattatore di memoria stato del listener* per memorizzare informazioni sullo stato, in un sistema MQSeries Everyplace o in una coda di sincr. del sistema MQSeries.

Il listener delle code di trasmissione consente alle code remote di MQSeries di fare riferimento alle code locali di MQSeries Everyplace. È inoltre possibile creare le code remote di MQSeries Everyplace che si riferiscono alle code locali di MQSeries. Queste code remote di MQSeries Everyplace vengono definite *code di MQSeries-bridge* e possono essere utilizzate per estrarre (get), inserire (put) e sfogliare (browse) i messaggi su una coda MQSeries.

## collegamento a MQSeries

Una coda MQSeries-bridge può contenere i seguenti attributi.

Tabella 23. MQSeries-bridge proprietà delle code

Proprietà	Spiegazione
Nomi alternativi (alias)	Nomi alternativi per la coda
Strumento di autenticazione	Deve essere nullo
Classe	Classe oggetto
Collegamento al client	Nome del servizio di collegamento al server da utilizzare
Compressore	Deve essere nullo
Programma di codifica	Deve essere nullo
Scadenza	Passato al trasformatore
Dimensione massima dei messaggi	Passato alla classe di regole
Modo	Deve essere sincrono
Proxy del programma di gestione code MQ	Nome del programma di gestione code di MQSeries a cui inviare per primo il messaggio
Collegamento MQSeries	Nome del collegamento per inviare il messaggio a MQSeries
Nome	Nome con cui MQSeries Everyplace identifica la coda remota MQSeries
Programma di gestione code corrente	Programma di gestione code che gestisce la definizione
Priorità	Priorità per i messaggi (se non sovrascritta da un valore di messaggio)
Nome coda remota di MQSeries	Nome della coda remota di MQSeries
Regola	Classe di regole utilizzata per le operazioni sulle code
Destinazione programma di gestione code	Programma di gestione code di MQSeries che contiene la coda
Trasformatore	Nome della classe del trasformatore che converte il messaggio dal formato MQSeries Everyplace al formato MQSeries
Tipo	Coda di collegamento MQSeries

**Nota:** Le classi del *programma di codifica*, dello *strumento di autenticazione* e del *compressore* definiscono un insieme di attributi di code che stabiliscono il livello di sicurezza per tutti i messaggi che arrivano sulla coda. Dal momento in cui il messaggio viene inviato da MQSeries Everyplace fino

a quando viene spostato alla coda di MQSeries-bridge, il messaggio è protetto almeno al livello di sicurezza delle code. Questi livelli di sicurezza *non* sono validi quando la coda MQSeries-bridge trasferisce il messaggio al sistema MQSeries durante il trasferimento vengono utilizzate le uscite di sicurezza, invio e ricezione nel collegamento al client. Non vengono eseguiti controlli per verificare che venga mantenuto il livello di sicurezza delle code.

Le code MQSeries-bridge sono solo asincrone. Le applicazioni asincrone devono quindi utilizzare una combinazione di code di archiviazione e home-server di MQSeries Everyplace, oppure una definizione di coda remota asincrona come passo intermedio quando vengono inviati dei messaggi alle code MQSeries-bridge.

Le applicazioni utilizzano le code di MQSeries-bridge come tutte le altre code remote di MQSeries Everyplace, utilizzando i metodi `putMessage`, `browseMessages` e `getMessage` della classe `MQQueueManager`. Il parametro del nome di coda di queste chiamate è il nome di coda di MQSeries-bridge, e il parametro del nome del programma di gestione code è il nome di MQSeries. Tuttavia, affinché il nome del programma di gestione code sia accettato dal server locale di MQSeries Everyplace, la definizione di collegamento al nome del programma di gestione code di MQSeries, deve esistere ed i parametri devono essere nulli, includendo il nome del canale.

**Nota:** esistono alcune limitazioni relative all'utilizzo di `getMessage` e `browseMessages` con le code di MQSeries-bridge. Non è possibile estrarre o sfogliare i messaggi dalle code di MQSeries-bridge che scelgono le definizioni di code remote di MQSeries. Non è inoltre possibile utilizzare get ID di conferma diversi da zero sulle code di MQSeries-bridge. L'operazione `getMessage` sulle code di MQSeries-bridge non fornisce una consegna sicura. Affinché tale operazione sia sicura, è necessario utilizzare listener di coda di trasmissione per trasferire i messaggi da MQSeries.

L'amministrazione di MQSeries-bridge viene svolta nello stesso modo di quella di un normale programma di gestione code di MQSeries Everyplace, utilizzando messaggi di tipo amministrativo. Vengono definite nuove classi di messaggi come adatte all'oggetto gestito. Tabella 13 a pagina 42 indica le classi di messaggi di amministrazione del gateway.

---

### Conversione messaggi

I messaggi di MQSeries Everyplace destinati a MQSeries passano attraverso il bridge per essere convertiti in formato MQSeries, utilizzando un trasformatore predefinito o uno specifico per la coda di destinazione. Un trasformatore personalizzato offre una maggiore flessibilità, ad esempio, è consigliabile

## collegamento a MQSeries

utilizzare una classe secondaria della classe di messaggi di MQSeries Everyplace per rappresentare i messaggi di tipo particolare sulla rete di MQSeries Everyplace. Nel gateway un trasformatore può convertire il messaggio in un formato MQSeries utilizzando qualsiasi corrispondenza adatta tra i campi e i valori di MQSeries, così come può aggiungere dati specifici per rappresentare il significato della classe secondaria.

Il trasformatore predefinito da MQSeries Everyplace a MQSeries non è in grado di sfruttare le informazioni della classe secondaria ma è stato progettato per una vasta gamma di situazioni. Presenta le caratteristiche seguenti:

- **Flusso dei messaggi da MQSeries Everyplace a MQSeries:**

Il trasformatore predefinito da MQSeries Everyplace a MQSeries funziona insieme alla classe MQeMQMsgObject. Questa classe è una rappresentazione di tutti i campi che è possibile trovare nell'intestazione del messaggio MQSeries. Utilizzando MQeMQMsgObject, l'applicazione può impostare i valori (come la priorità) utilizzando i metodi stabiliti. Quindi, quando MQeMQMsgObject (o un oggetto che deriva dalla classe MQeMQMsgObject) passa attraverso il trasformatore predefinito di MQSeries (MQeBaseTransformer) riceve i valori da MQeMSMsgObject e imposta i valori corrispondenti nel messaggio di MQSeries (per esempio, il valore di priorità viene copiato nel messaggio di MQSeries).

Se il messaggio trasferito non è un MQeMQMsgObject, e non deriva dalla classe MQeMQMsgObject, l'intero messaggio di MQSeries Everyplace viene copiato nel corpo del messaggio di MQSeries (*incanalato*). Il campo del formato messaggio nell'intestazione del messaggio di MQSeries viene impostato per indicare che il messaggio MQSeries contiene un messaggio in formato "incanalato" MQSeries Everyplace.

- **Flusso di messaggi da MQSeries a MQSeries Everyplace:**

I messaggi di MQSeries per MQSeries Everyplace sono gestiti in modo simile a quelli che si spostano nella direzione opposta. Il trasformatore predefinito esamina il campo del tipo di messaggio dell'intestazione di MQSeries e agisce di conseguenza.

Se l'intestazione MQSeries indica un messaggio "incanalato" MQSeries Everyplace, il corpo del messaggio di MQSeries viene ricostruito come il messaggio MQSeries Everyplace originale e viene inviato poi sulla rete MQSeries Everyplace.

Se invece non si tratta di un messaggio "incanalato" MQSeries Everyplace, si estrae il contenuto l'intestazione del messaggio MQSeries e si memorizza nell'oggetto MQeMQMsgObject. Il corpo del messaggio MQSeries viene considerato come campo di byte e inserito nell'oggetto MQeMQMsgObject. MQeMQMsgObject viene quindi inviato alla rete MQSeries Everyplace.

Il funzionamento di questa classe MQeMQMsgObject e del trasformatore predefinito indicano:

- Un messaggio MQSeries Everyplace può essere inviato da una rete MQSeries a una rete MQSeries Everyplace senza modifiche.
- Un messaggio MQSeries può essere inviato da una rete MQSeries Everyplace a una rete MQSeries senza modifiche.
- Un'applicazione MQSeries Everyplace può guidare qualsiasi altra applicazione MQSeries esistente senza alcuna modifica all'applicazione MQSeries.

---

### Funzione

Le code remote MQSeries sono abilitate per operazioni di messaggistica sincrona di MQSeries Everyplace, da un programma di gestione code MQSeries Everyplace; tutte le altre operazioni simili devono essere asincrone.

Non è possibile inviare messaggi di amministrazione MQSeries Everyplace ad un programma di gestione code di MQSeries. AdminQ non esiste e il formato del messaggio di amministrazione è diverso da quello utilizzato da MQSeries.

---

### Compatibilità

Una rete MQSeries Everyplace può esistere indipendentemente da MQSeries, ma in molte situazioni sono necessari entrambi i prodotti per rispondere ai requisiti dell'applicazione. MQSeries Everyplace può essere integrato in una rete MQSeries esistente con compatibilità, compresi gli aspetti elencati sotto:

#### **indirizzamento e denominazione:**

- semantica di indirizzamento uguale utilizzando un indirizzo di coda o un programma di gestione code
- Utilizzo comune di uno spazio nome ASCII

#### **Applicazioni:**

MQSeries Everyplace è in grado di supportare le applicazioni di MQSeries esistenti senza apportarvi modifiche.

#### **Canali:**

I gateway di MQSeries Everyplace utilizzano canali client di MQSeries.

#### **Interscambio e contenuto dei messaggi:**

- interscambio di messaggi tra MQSeries Everyplace e MQSeries
- rete messaggi invisibile: i messaggi da MQSeries Everyplace o MQSeries possono viaggiare sull'altra rete senza alcuna modifica
- supporto reciproco per campi identificati nell'intestazione del messaggio MQSeries

## collegamento a MQSeries

- consegna di messaggi assicurata per una sola volta

MQSeries Everyplace Versione 1.2 non supporta tutte le funzioni di MQSeries. A parte le considerazioni sull'ambiente, il sistema operativo e le comunicazioni, alcune delle differenze più significative sono ampiamente descritte di seguito. Tuttavia all'interno di MQSeries Everyplace è possibile svolgere molte delle attività di applicazioni con strumenti alternativi, utilizzando le funzioni di MQSeries Everyplace, oppure sfruttando le classi secondarie, sostituendo le classi fornite oppure utilizzando regole, interfacce e altre funzioni di personalizzazione all'interno del prodotto.

- Nessun supporto cluster
- Nessun supporto elenco di distribuzione
- Nessun messaggio raggruppato o segmentato
- Nessuna funzione di attesa o di bilanciamento carico del lavoro
- Nessun messaggio di riferimento
- Nessuna opzione di relazione
- Nessun supporto per code condivise
- Nessun errore
- Nessun supporto per unità di lavoro, nessuna coordinazione XA

Le caratteristiche di scalabilità e prestazioni sono diverse.

---

### Consegna assicurata

Anche se MQSeries Everyplace e MQSeries offrono entrambi la consegna assicurata, ciascuno fornisce diversi livelli di assicurazione. Quando un messaggio viene inviato da MQSeries Everyplace a MQSeries, il trasferimento è assicurato solo se si utilizza la combinazione di funzioni `putMessage` e `confirmPutMessage` (vedere "Configurazione del programma di gestione code" a pagina 34). Quando un messaggio viene inviato da MQSeries a MQSeries Everyplace, il trasferimento è assicurato solo se il messaggio di MQSeries è definito permanente.

---

## Capitolo 7. Interfacce di programmazione

*MQSeries Everyplace SPI (Systems Programming Interface)* è l'interfaccia di programmazione per MQSeries Everyplace. Vengono supportati due linguaggi, Java e C:

**La versione Java** fornisce accesso a tutte le funzioni di MQSeries Everyplace. Le classi, i metodi e le procedure predefinite sono descritte in dettaglio in *MQSeries Everyplace for Multiplatforms Programming Reference*; esempi di programmazione di MQSeries Everyplace sono forniti in *MQSeries Everyplace for Multiplatforms Programming Guide*.

**Il supporto C per Palm** fornisce l'accesso ai sottogruppi della funzione di MQSeries Everyplace per poter utilizzare i dispositivi Palm. Vengono forniti dettagli su queste classi, procedure e informazioni sulla programmazione in *MQSeries Everyplace Native Client Information*



---

## Capitolo 8. Introduzione a MQSeries Everyplace

MQSeries Everyplace è una famiglia di prodotti che fornisce gli strumenti necessari allo sviluppo, all'installazione e alla gestione della messaggistica e dell'accodamento di MQSeries Everyplace. La famiglia si adatta a:

1. Il prodotto fornito di licenza di *MQSeries Everyplace* (disponibile su supporti fisici o scaricandolo dal sito Web all'indirizzo <http://software.boulder.ibm.com/dl/mqsem/mqsem-p> ). I prodotti forniti di licenza includono:
  - Classi Java di MQSeries Everyplace
  - Classi degli strumenti di aiuto
  - Esempi di codici sorgente di applicazione
  - Programmi di utilità
  - Manuali di riferimento
  - Informazioni relative alle licenze

Il supporto fisico Program Product concede inoltre l'utilizzo del prodotto per un uso produttivo su alcune piattaforme. Ulteriori unità di capacità devono essere acquistate nel caso si vogliano utilizzare macchine più grandi o che utilizzano MQSeries-bridge.

2. SupportPacs di MQSeries Everyplace (Disponibile se scaricato dal sito Web all'indirizzo <http://software.boulder.ibm.com/dl/mqsem/mqsem-p> (come sopra) oppure all'indirizzo <http://www.ibm.com/software/mqseries/everyplace>). Questi componenti sono integrazioni essenziali del prodotto ed includono:

**EAP1: MQSeries Everyplace - Device code for the Palm OS**

Supporto del linguaggio di programmazione C per lo sviluppo dell'applicazione della Versione 1.0.1 di MQSeries Everyplace su Palm OS. Questo codice è incluso sul CD del prodotto contenuto nel file eap1.zip

**EP01: MQSeries Everyplace - Performance Report**

Analizza le prestazioni MQSeries Everyplace su varie piattaforme client

**ES01: MQSeries Everyplace - Administration Tool (MQExplorer v1.0)**

Uno strumento generico per tutte le piattaforme Java che consentono la gestione grafica dei programmi di gestione code MQSeries Everyplace

### ES02: MQSeries Everyplace - Explorer (MQe\_Explorer v1.2)

Uno strumento di gestione MQSeries Everyplace sviluppato esclusivamente per supportare la gamma dei sistemi operativi Microsoft Windows

Gli strumenti di gestione del SupportPacs di MQSeries Everyplace giocano un ruolo importante in tutte le fasi di sviluppo e rollout delle applicazioni. Sono più sofisticati dei programmi di utilità forniti con i prodotti e sono un valido aiuto per configurare, per analizzare le reti principali e gestire i sistemi di produzione.

---

## Uso di MQSeries Everyplace

Data la varietà di utenti di MQSeries Everyplace, il prodotto non viene installato, configurato ed analizzato nel modo in cui vengono effettuate le stesse operazioni sugli altri membri della famiglia di MQSeries. Il concetto qui sottolineato è che esistono tre fasi nell'adozione di MQSeries Everyplace da parte dell'impresa:

### 1. Fase di sviluppo e di creazioni di un prototipo

Nella fase iniziale di apprendimento, di sviluppo e di creazione di un prototipo, il prodotto MQSeries Everyplace può essere installato senza costi aggiuntivi in base agli accordi di licenza IBM di tale prodotto. Le applicazioni MQSeries Everyplace vengono sviluppate utilizzando le funzioni delle classi Java e delle routine C di MQSeries Everyplace. Queste applicazioni possono essere assemblate in diversi modi:

- Un programma di gestione code MQSeries Everyplace può essere impostato come un daemon con una o più applicazioni avviate nella stessa macchina virtuale Java e che condivide un programma di gestione code comune.
- L'applicazione incorpora le classi richieste di MQSeries Everyplace in modo tale che l'applicazione eseguita sulla macchina nella quale non è stata installata MQSeries Everyplace, avvia il proprio programma di gestione code nel proprio JVM.
- L'applicazione utilizza le classi MQSeries Everyplace esistenti sulla macchina di destinazione.

Il supporto da parte dell'IBM non è incluso nella licenza relativa allo sviluppo. Tuttavia, il supporto durante lo sviluppo dell'applicazione ed oltre viene fornito insieme alla licenza relativa all'installazione (consultare sotto).

### 2. Fase di analisi

La fase di sviluppo corrisponde al rollout e all'utilizzo delle applicazioni sviluppate, quindi, in conformità ai termini di licenza dell'IBM di MQSeries Everyplace, vengono richieste le unità di capacità per poter utilizzare il prodotto. Le classi possono essere

distribuite solamente utilizzando l'applicazione conforme ai termini di licenza dell'IBM. In caso contrario, gli utenti devono fornire le classi necessarie.

### 3. Fase di gestione

Di conseguenza, quando i programmi di gestione code di MQSeries Everyplace sono attivati all'interno di una rete, gli strumenti diventano necessari per la gestione ed il controllo. Il supporto per MQSeries Everyplace viene fornito secondo i termini dell'Accordo internazionale di licenza per i programmi.

La fase di adozione giustifica la variazione di livello di supporto delle piattaforme. Per MQSeries Everyplace con unità di capacità (e SupportPacs di categoria 3), l'IBM effettua le seguenti distinzioni:

- Piattaforme in cui vengono supportate sia l'installazione che l'applicazione:
  - saranno riconosciute documentazioni sui problemi relativi all'utilizzo, all'installazione e/o allo sviluppo dell'applicazione
- Piattaforme in cui viene permesso lo sviluppo dell'applicazione ma non supportato direttamente:
  - saranno richieste documentazioni da riprodurre sulla piattaforma supportata
- Piattaforme in cui viene supportato lo sviluppo dell'applicazione:
  - saranno accettate documentazioni relative allo sviluppo dell'applicazione

---

## Come acquisire esperienza

Esistono molti modi di ottenere informazioni relative all'utilizzo di MQSeries Everyplace. L'esperienza suggerisce che, il modo più semplice di utilizzo del prodotto è l'avvio di un programma di gestione code, seguito da una rete semplice di MQSeries Everyplace. Per cui, la scrittura di una semplice applicazione, può preparare ad uno studio più approfondito del prodotto. Nella prima fase, può non essere di alcun aiuto nell'esaminare gli altri membri della famiglia di MQSeries. Successivamente, quando la funzionalità del collegamento è rilevante, diventa essenziale comprenderne la procedura.

Utilizzando tale strategia, ai nuovi utenti viene raccomandato di seguire il seguente metodo:

Comprendere gli elementi essenziali forniti in *Introduzione a MQSeries Everyplace for Multiplatforms* (questa pubblicazione) ed effettuare le operazioni riportate di seguito:

1. Se si utilizza una macchina il cui sistema operativo è Windows, scaricare SupportPac ES02 (MQe\_Explorer v 1.2) di MQe\_Explorer. Il prodotto con licenza può non essere installato, in questo caso, l'utilizzo dello sviluppo deve essere conforme ai termini di licenza.

2. Seguire le istruzioni riportate in “Utilizzo di ES02: MQe\_Explorer” .

### Utilizzo di ES02: MQe\_Explorer

MQe\_Explorer viene fornita di una guida per l'utente completa che descrive la maniera più veloce di configurazione dei programmi di gestione code. Il manuale viene raccomandato sia agli utenti che agli amministratori di MQe\_Explorer, poiché include vari script di esempio che illustrano le più importanti funzioni di MQSeries Everyplace. SupportPac include due versioni eseguibili di MQe\_Explorer:

#### MQe\_ExplorerX.exe

Questa versione, contiene all'interno del file .exe tutte le classi di MQSeries Everyplace necessarie. MQe\_ExplorerX.exe può essere eseguita anche se MQSeries Everyplace non è stata installata. È adatta ai principianti.

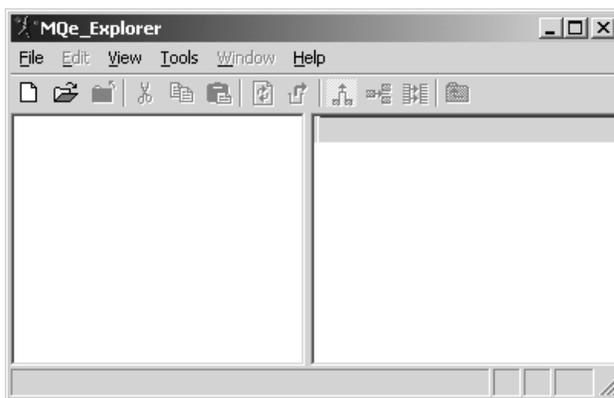
#### MQe\_Explorer.exe

Questa versione dipende dalla MQSeries Everyplace precedentemente installata. La versione è in grado di scegliere il livello più aggiornato della libreria di MQSeries Everyplace, ed è di dimensioni più ridotte. È adatta ai programmatori e agli amministratori.

È possibile creare dei programmi di gestione code facilmente, le seguenti istruzioni mostrano le capacità di MQSeries Everyplace.

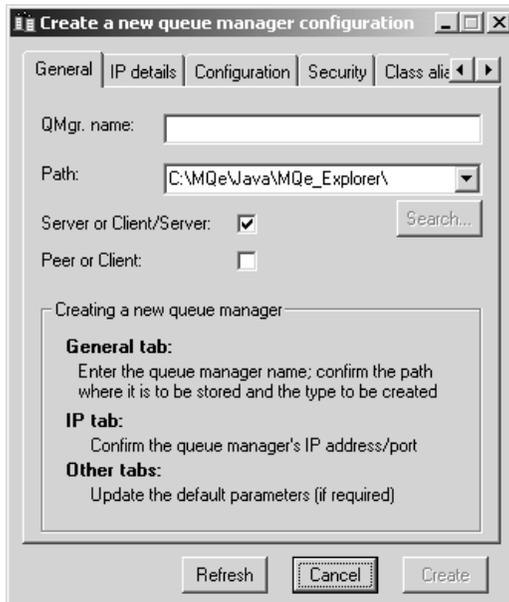
1. Fare doppio clic sull'icona MQe\_ExplorerX.exe  .

Un messaggio indica che non sono state trovate le opzioni salvate, fare clic su **OK** (questo messaggio non verrà più visualizzato). Viene visualizzata la seguente finestra di amministrazione:

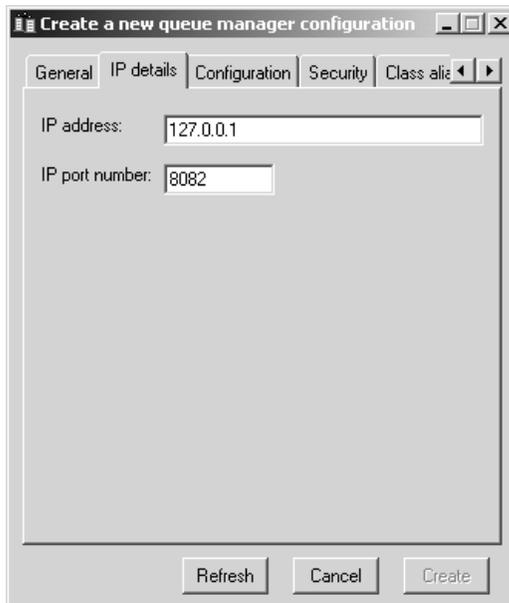


2. Fare clic sulla nuova icona  sulla barra degli strumenti.

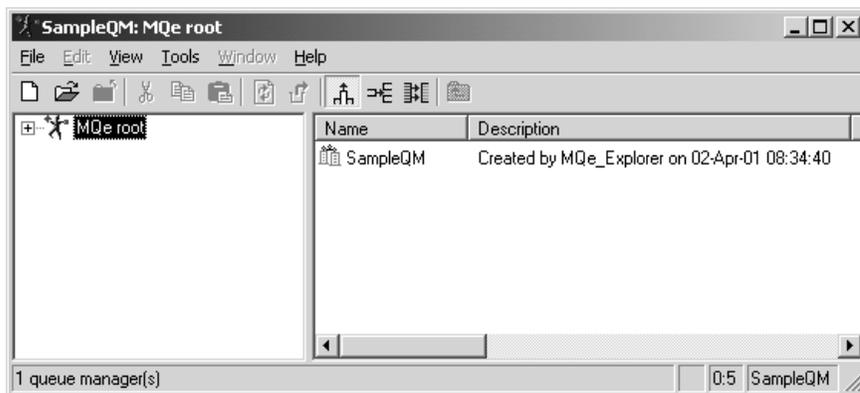
Questa icona crea un nuovo programma di gestione code. Viene visualizzata la seguente finestra:



3.
  - a. Digitare il nome del programma di gestione code (ad esempio, SampleQM).
  - b. Selezionare la scheda **IP details**

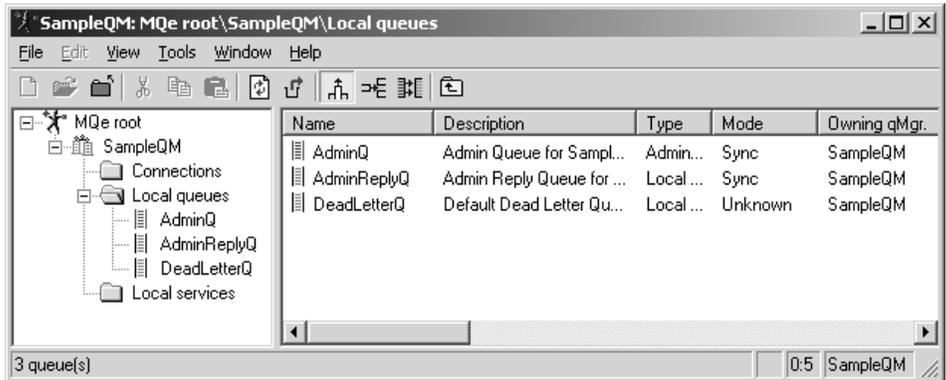


4.
  - a. Digitare un indirizzo IP - l'indirizzo è quello della macchina utilizzata; un valore da digitare a questo punto è 127.0.0.1 (host locale).
  - b. Fare clic sul pulsante **Crea**. Viene visualizzato un messaggio che identifica il file di inizializzazione creato (il nome è necessario per l'accesso al programma di gestione code);
  - c. Fare clic su **OK**, viene visualizzata la seguente finestra:



5. Viene creato un programma di gestione code del server eseguito in JVM. È in collegamento con la porta 8081 per le richieste di connessione del canale client/server. Se i simboli + sono espansi nella struttura ad albero nel

pannello di sinistra e sia il pannello di sinistra che la finestra sono ridimensionati, ciò indica che sono state create quattro code.



6. Per effettuare ulteriori esperimenti, seguire le istruzioni della *MQe\_Explorer User Guide*. È possibile creare code, collegamenti, messaggi ed anche reti complete di MQSeries Everyplace. Le applicazioni possono essere caricate nello stesso programma di gestione code ed eseguito in contemporanea con MQe\_Explorer.



---

## Appendice. Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti. È possibile che negli altri paesi l'IBM non offra i prodotti, i servizi o le funzioni illustrati in questo documento. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti dall'IBM, possono essere usati prodotti, programmi o servizi funzionalmente equivalenti che non comportino la violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. È responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri programmi e/o prodotti, fatta eccezione per quelli espressamente indicati dall'IBM.

L'IBM può avere brevetti o domande di brevetto in corso relativi a quanto trattato nella presente pubblicazione. La fornitura di questa pubblicazione non implica la concessione di alcuna licenza su di essi. Chi desiderasse ricevere informazioni relative alle licenze può rivolgersi per iscritto a:

Director of Commercial Relations IBM Europe  
Schoenaicher Str. 220  
D-7030 Boeblingen  
Deutschland

**Il seguente paragrafo non è valido per il Regno Unito o per tutti i paesi le cui leggi nazionali siano in contrasto con le disposizioni in esso contenute:**  
L'INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE QUESTA PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA", SENZA ALCUNA GARANZIA, ESPLICITA O IMPLICITA, IVI INCLUSE EVENTUALI GARANZIE DI COMMERCIALIZZATA ED IDONEITÀ AD UNO SCOPO PARTICOLARE. Alcune stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni; quindi la presente dichiarazione potrebbe essere non essere a voi applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche verranno incorporate nelle nuove edizioni della pubblicazione. L'IBM si riserva il diritto di apportare miglioramenti e/o modifiche al prodotto o al programma descritto nel manuale in qualsiasi momento e senza preavviso.

## informazioni particolari

Tutti i riferimenti a siti Web non dell'IBM contenuti in questo documento sono forniti solo per consultazione. I materiali disponibile presso i siti Web non fanno parte di questo prodotto e l'utilizzo di questi è a discrezione dell'utente.

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente dall'IBM e dalla Selfin e diventeranno esclusiva delle stesse.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

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

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento di un addebito.

Il programma su licenza descritto in questo manuale e tutto il materiale su licenza ad esso relativo sono forniti dall'IBM nel rispetto delle condizioni previste dalla licenza d'uso.

---

### Marchi

I seguenti termini sono marchi della International Business machines Corporation.

AIX    AS/400    IBM    MQSeries    OS/390

Microsoft, Windows, Windows NT e i logo Windows sono marchi della Microsoft Corporation.

Java e tutti i marchi basati su Java sono logo e marchi della Sun Microsystems, Inc..

Altri nomi di società, prodotti e servizi possono essere marchi di altri.



---

## Glossario

Questo glossario descrive i termini utilizzati in questa pubblicazione e le parole che hanno un significato diverso da quello colloquiale. In alcuni casi una definizione può non essere l'unica riferita ad un termine, ma rende il significato particolare con cui questa parola viene utilizzata in questa pubblicazione.

Se non si riesce a trovare un termine, consultare l'indice oppure *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

**accodamento messaggi.** Una tecnica di programmazione con cui ciascun programma di un'applicazione comunica con gli altri programmi posizionando i messaggi su delle code (accodamento).

**API (Application Programming Interface).** Un'API è costituita da funzioni e variabili che i programmatori possono utilizzare per le loro applicazioni.

**autenticatore.** Un programma che controlla gli utenti che inviano e ricevono i messaggi.

**bridge.** Un oggetto di MQSeries Everyplace che consente di inviare messaggi tra MQSeries Everyplace e altri sistemi di messaggistica, compreso MQSeries.

**browser Web.** Un programma che elabora e visualizza informazioni distribuite sul World Wide Web.

**canale.** Vedere *canale dinamico* e *canale MQI*.

**canale dinamico.** Un canale dinamico connette i dispositivi di MQSeries Everyplace, trasferisce i messaggi sincroni e asincroni e risponde in modo bidirezionale.

**canale MQI.** Un canale MQI collega un client MQSeries a un programma di gestione code su un server, trasferisce chiamate MQI e risponde in modo bidirezionale.

**classe.** Una classe è una raccolta generale di dati e metodi per operare sui dati. Una classe può essere istanziata per produrre un oggetto che costituisce l'istanza della classe.

**classe principale.** Una classe principale è una classe ampliata in altre classi. I metodi pubblico e protetto e le variabili di una classe principale sono disponibili per la classe secondaria.

**classe secondaria.** Una classe secondaria è una classe che costituisce un ampliamento della classe principale. La classe secondaria eredita i metodi pubblico e protetto, oltre alle variabili della classe principale di appartenenza.

**client.** In MQSeries, un client è un componente runtime che consente di accedere ai servizi di accodamento su un server per applicazioni per utenti in locale.

**coda.** Una coda è un oggetto di MQSeries. Le applicazioni di accodamento messaggi possono mettere o ricevere messaggi su una coda

**coda di messaggi.** Vedere *coda*

**compressore.** Un programma che comprime un messaggio per ridurre il volume di dati da trasmettere.

**gateway.** Un gateway MQSeries Everyplace gateway (o server) è un computer che esegue il codice MQSeries Everyplace e comprende un programma di gestione canali.

**HTML (Hypertext Markup Language).** Un linguaggio utilizzato per definire informazioni da visualizzare sul World Wide Web.

**incapsulamento.** L'incapsulamento è una tecnica di programmazione orientata all'oggetto

che protegge i dati di un oggetto e consente ai programmatori di accedere e manipolare i dati solo con richiami al metodo.

**interfaccia.** Un'interfaccia è una classe contenente solo metodi astratti, senza variabili di istanza. Un'interfaccia fornisce una serie di metodi diffusi da implementare con classi secondarie di alcune classi diverse.

**Internet.** Internet è una rete pubblica cooperativa di informazioni condivise. Fisicamente Internet utilizza una serie secondaria delle risorse totali di tutte le reti di telecomunicazioni pubbliche attualmente esistenti. Tecnicamente, ciò che caratterizza Internet come rete pubblica cooperativa è l'utilizzo di una serie di protocolli definiti TCP/IP (Transport Control Protocol/Internet Protocol).

**istanza.** Un'istanza è un oggetto. Quando una classe viene istanziata per produrre un oggetto, si dice che l'oggetto è un'istanza della classe.

**JDK (Java Developers Kit).** Un pacchetto software distribuito dalla Sun Microsystems per sviluppatori Java, che comprende il programma di interpretazione Java, classi e strumenti di sviluppo Java: compilatore, debugger, disassembler, appletviewer, generatore di file stub e generatore di documentazione.

**JNDI (Java Naming and Directory Service).** Un'API specificata nel linguaggio di programmazione Java. Fornisce funzioni di denominazione e directory per applicazioni scritte con linguaggio di programmazione Java.

**LDAP (Lightweight Directory Access Protocol).** LDAP è un protocollo client-server per avere accesso a un servizio di directory.

**messaggio.** Nelle applicazioni di accodamento messaggi, un messaggio è una comunicazione spedita tra programmi.

**messaggistica asincrona.** Un metodo di comunicazione tra programmi in cui i programmi stessi inseriscono i messaggi nelle code di messaggi. Con la messaggistica asincrona, il programma di invio funziona senza

dover aspettare una risposta al messaggio. E' opposto alla *messaggistica sincrona*.

**messaggistica sincrona.** Un metodo di comunicazione tra programmi in cui i programmi stessi inseriscono i messaggi nelle code di messaggi. Con la messaggistica sincrona, il programma di invio deve aspettare una risposta al messaggio prima di riprendere il funzionamento. E' opposto alla *messaggistica asincrona*.

**metodo.** Il metodo è un termine di programmazione orientato all'oggetto per una funzione o una procedura.

**MQSeries.** MQSeries è una famiglia di programmi con licenza IBM che forniscono servizi di accodamento messaggi.

**oggetto.** (1) In Java, un oggetto è un'istanza di una classe. Una classe modella un gruppo di oggetti; un oggetto modella un determinato membro del gruppo. (2) In MQSeries, un oggetto è un programma di gestione code, una coda o un canale.

**pacchetto.** Un pacchetto in Java consente a una parte di codice Java di accedere a una serie specifica di classi. Il codice Java che costituisce un determinato pacchetto ha accesso a tutte le classi del pacchetto e a tutti i metodi e i campi non protetti delle classi.

**PDA (personal digital assistant).** Un personal computer di piccole dimensioni.

**privato.** Un campo privato non può essere visualizzato al di fuori della propria classe.

**programma di codifica.** Un programma che codifica un messaggio per garantire sicurezza durante la trasmissione.

**programma di gestione canali.** Un oggetto di MQSeries Everyplace che supporta i canali di comunicazione logica a più utenti tra gli utenti finali.

**programma di gestione code.** Un programma di gestione code fornisce alle applicazioni dei servizi di accodamento messaggi.

**protetto.** Un campo protetto può essere visualizzato solo all'interno della propria classe, in una classe secondaria o in pacchetti di cui la classe fa parte

**pubblico.** Una classe o un'interfaccia pubblica può essere visualizzata sempre. Un metodo o una variabile pubblica è visualizzabile in tutti i luoghi in cui può essere visualizzata la relativa classe

**server.** (1) Un MQSeries Everyplace server è un dispositivo dotato di un programma di gestione canali di MQSeries Everyplace. (2) Un server MQSeries è un programma di gestione code che fornisce servizi di accodamento messaggi alle applicazioni client in esecuzione su una stazione di lavoro remota. (3) Più in generale, un server è un programma che risponde alle richieste di informazione nel modello particolare per il flusso di informazioni a due programmi per client/server. (3) Il computer su cui viene eseguito un programma per server.

**servlet.** Un programma Java concepito per eseguire unicamente un server Web.

**TCP/IP (Transmission Control Protocol/Internet Protocol).** Una serie di protocolli di comunicazione che supporta funzioni di connessione peer-to-peer per reti di aree locali ed estese.

**Web.** Vedere World Wide Web.

**Web (World Wide Web).** Il World Wide Web è un servizio Internet, basato su una serie di protocolli comuni, che consente ad un server configurato in modo particolare di distribuire documenti su Internet secondo una procedura standard.



---

## Bibliografia

Pubblicazioni correlate:

- *MQSeries Everyplace for Multiplatforms - Readme*, GC34-5862
- *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 Quick Beginnings*, GC34-5389-00



---

## Indice analitico

### A

a chi si rivolge questa pubblicazione v  
adattatori, MQSeries Everyplace 46  
adattatori MQSeries Everyplace 46  
ambienti, software 3  
ambienti software 3  
amministrazione con MQSeries Everyplace 41  
amministrazione MQSeries Everyplace 41  
applicazioni, caricamento 61  
applicazioni, MQSeries Everyplace 14  
applicazioni di MQSeries Everyplace 14  
asincrona, messaggistica 47  
autoregistrazione 54

### B

bridge, MQSeries 63

### C

canali 9  
canali, client 17  
canali, dinamici 17, 44  
canali client 17  
canali client-server 9  
canali dinamici 9, 17, 44  
canali peer-to-peer 9  
caratteristiche 13  
caricamento applicazioni 61  
classi, MQSeries Everyplace 60  
Classi di MQSeries Everyplace 60  
client, MQSeries 8  
client MQSeries 8  
cliente, requisiti 14  
code, bridge MQSeries 27, 28  
code, memorizzazione e inoltro 26  
code, MQSeries Everyplace 24  
code bridge di MQSeries 28  
code dell'home server 27  
code di bridge di MQSeries 27  
code di memorizzazione e inoltro 26  
code di MQSeries Everyplace 24  
code locali 24  
code remote 25  
codifica 48  
collegamento, client-server 59

collegamento, peer-to-peer 59  
collegamento client-server 59  
collegamento peer-to-peer 59  
compatibilità con MQSeries 73  
compressione 48  
comunicazioni 58  
configurazione 56  
configurazioni, esempio 10  
configurazioni di esempio 10  
consegna di messaggi assicurata 74  
consegna messaggi, assicurata 74  
contenuti del prodotto 17  
conversione messaggi 71

### D

descrizione 1  
destinatari v  
Dispositivi, MQSeries Everyplace 17  
dispositivi MQSeries Everyplace 17

### E

entità, autenticabili 54  
entità autenticabili 54

### F

famiglia MQSeries 7  
formato dei dati dump 23

### G

gateway, MQSeries Everyplace 17  
gateway MQSeries Everyplace 17  
gestione dei collegamenti dialup 46  
gestori di code 8, 9

### H

home server, MQSeries Everyplace 27

### I

informazioni, legali 85  
informazioni legali 85  
informazioni su questa pubblicazione v  
interfacce, programmazione 75  
interfacce di programmazione 75  
interfaccia, sicurezza 56  
interfaccia di sicurezza 56  
interfaccia per MQSeries 63

### L

lettori v  
listener di canali 58

locali, code 24

### M

marchi 87  
messaggi, amministrazione 41  
messaggi, oggetti 18  
messaggi di amministrazione 41  
messaggistica, MQSeries 7  
messaggistica asincrona 47  
messaggistica di MQSeries 7  
messaggistica diffusa vi, 8  
messaggistica distribuita vi, 8  
messaggistica host vi, 8  
messaggistica sincrona 48  
mini-certificati 54  
MQeAttribute 52  
MQeMTrustAttribute 52  
MQSeries, compatibilità con 73  
MQSeries, interfaccia per 63  
MQSeries-bridge 9, 63  
MQSeries Integrator vi, 7  
MQSeries Workflow vi, 7

### O

oggetti, MQSeries Everyplace 18  
oggetti di MQSeries Everyplace 18  
oggetti messaggi 18  
oggetto bridge 65  
oggetto collegamento 66  
oggetto listener 69, 70  
oggetto proxy del programma di gestione code 66  
operazioni, programma di gestione code 38  
operazioni dei programmi di gestione code 38

### P

panoramica 1  
personalizzazione 56  
prerequisiti v, 3  
prodotto, contenuti 17  
programma di controllo 44  
programma di gestione canali 58  
programma di gestione code 58  
programmi di gestione code, MQSeries Everyplace 31  
programmi di gestione code di MQSeries Everyplace 31

## R

- registrazioni eventi 47
- registro 53
- registro, MQSeries Everyplace 17
- registro, privato 54
- registro, pubblico 55
- registro di MQSeries Everyplace 53
- registro MQSeries Everyplace 17
- registro privato 54
- registro pubblico 55
- regole, MQSeries Everyplace 56
- regole del programma di gestione code 58
- regole di attributo 57
- regole di bridge MQSeries 57
- regole di coda 57
- regole di MQSeries Everyplace 56
- remote, code 25
- replica certificati 55
- replica di certificati 55
- requisiti del cliente 14
- reti, MQSeries 63
- reti, MQSeries Everyplace 47, 63
- Reti MQSeries 63
- Reti MQSeries Everyplace 47, 63

## S

- server, MQSeries 8
- server MQSeries 8
- servizio di creazione
  - mini-certificati 55
- servizio di creazione mini-certificati
  - predefiniti 55
- sicurezza, a livello di messaggio 52
- sicurezza, basata sulle code 50
- sicurezza, locale 50
- sicurezza, MQSeries Everyplace 48
- sicurezza a livello di messaggio 52
- sicurezza basata sulle code 50
- sicurezza di MQSeries
  - Everyplace 48
- sicurezza locale 50
- sincrona, messaggistica 48
- sistemi operativi, supportati 3
- sistemi operativi richiesti 3
- sistemi operativi supportati 3
- SPI 75
- stili di collegamento 58
- stili di collegamento, multipli 59
- stili di collegamento multipli 59

## T

- termini vi
- traccia di MQSeries Everyplace 46
- trasformatori 71

---

# Riservato ai commenti del lettore

**MQSeries Everyplace for Multiplatforms**  
**Introduzione**  
**Versione 1.2**

**Pubblicazione N. GC13-2908-02**

Commenti relativi alla pubblicazione in oggetto potranno contribuire a migliorarla. Sono graditi commenti pertinenti alle informazioni contenute in questo manuale ed al modo in cui esse sono presentate. Si invita il lettore ad usare lo spazio sottostante citando, ove possibile, i riferimenti alla pagina ed al paragrafo.

Si prega di non utilizzare questo foglio per richiedere informazioni tecniche su sistemi, programmi o pubblicazioni e/o per richiedere informazioni di carattere generale.

Per tali esigenze si consiglia di rivolgersi al punto di vendita autorizzato o alla filiale IBM della propria zona oppure di chiamare il "Supporto Clienti" IBM al numero verde 167-017001.

I suggerimenti ed i commenti inviati potranno essere usati liberamente dall'IBM e dalla Selfin e diventeranno proprietà esclusiva delle stesse.

Commenti:

Si ringrazia per la collaborazione.

Per inviare i commenti è possibile utilizzare uno dei seguenti modi.

- Spedire questo modulo all'indirizzo indicato sul retro.
- Inviare un fax al numero: +39-081-660236
- Spedire una nota via email a: [translationassurance@selfin.it](mailto:translationassurance@selfin.it)

Se è gradita una risposta dalla Selfin, si prega di fornire le informazioni che seguono:

---

Nome

---

Indirizzo

---

Società

---

Numero di telefono

---

Indirizzo e-mail

Indicandoci i Suoi dati, Lei avrà l'opportunità di ottenere dal responsabile del Servizio di Translation Assurance della Selfin S.p.A. le risposte ai quesiti o alle richieste di informazioni che vorrà sottoporci. I Suoi dati saranno trattati nel rispetto di quanto stabilito dalla legge 31 dicembre 1996, n.675 sulla "Tutela delle persone e di altri soggetti rispetto al trattamento di dati personali". I Suoi dati non saranno oggetto di comunicazione o di diffusione a terzi; essi saranno utilizzati "una tantum" e saranno conservati per il tempo strettamente necessario al loro utilizzo.



Selfin S.p.A.  
Translation Assurance

Via F. Giordani, 7

80122 NAPOLI





Printed in Denmark by IBM Danmark A/S

GC13-2908-02

