

MQSeries Everyplace



Introduction

Version 1

MQSeries Everyplace



Introduction

Version 1

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à l'«Annexe. Remarques» à la page 61

Licence MQSeries Everyplace version 1 est un ensemble d'outils de développement qui permet aux utilisateurs d'écrire des applications MQSeries Everyplace et de créer un environnement dans lequel les exécuter.

Les conditions de licence à l'achat de ces outils déterminent l'environnement dans lequel ils peuvent être utilisés.

Si MQSeries Everyplace a été acquis pour une exploitation en tant qu'**unité** (client), il **ne peut pas** être utilisé pour créer un **gestionnaire de canal MQSeries Everyplace** ou un **programme d'écoute de canal MQSeries Everyplace**.

La présence d'un **gestionnaire de canal MQSeries Everyplace** ou d'un **programme d'écoute de canal MQSeries Everyplace** définit un environnement **passerelle** (serveur), qui requiert une licence de passerelle.

Première édition – juin 2000

Réf. US : GC34-5843-00

Cette édition s'applique à MQSeries Everyplace version 1.0 et à toutes les versions et modifications ultérieures, sauf spécification contraire dans les nouvelles éditions.

LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT". IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

Compagnie IBM France
Direction Qualité
Tour Descartes
92066 Paris-La Défense Cedex 50

© Copyright IBM France 2000. Tous droits réservés.

© Copyright International Business Machines Corporation 2000. All rights reserved.

Table des matières

Figures	v
Tableaux.	vii
A propos de ce manuel.	ix
Lecteurs concernés	ix
Connaissances requises	ix
Terminologie	ix
Chapitre 1. Généralités	1
Remarques sur la version 1.0	1
Chapitre 2. Conditions préalables	3
Chapitre 3. La famille MQSeries	5
Chapitre 4. Conditions requises	11
Capacités	11
Applications	11
Exigences des clients	12
Chapitre 5. Concepts du produit	13
Introduction	13
Objets message	14
Format des données de vidage	18
Files d'attente	18
Gestionnaire de files d'attente	22
Opérations du gestionnaire de files d'attente	24
Administration	26
Messages d'administration	26
Administration sélective	28
Surveillance et actions connexes	29
Canaux dynamiques	29
Adaptateurs	30
Gestion de connexion commutée	30
Trace	31
Journal des événements	31
Réseau MQSeries Everyplace	31
Configurations et évolutivité	31
Livraison asynchrone des messages	33
Livraison synchrone des messages	34
Sécurité	34
Sécurité locale MQSeries Everyplace	35
Sécurité MQSeries Everyplace au niveau des files d'attente	36
Sécurité au niveau message	37
Registre	38
Entités MQSeries Everyplace authentifiables	39

Registre privé et accréditations	39
Enregistrement automatique	40
Registre public et réplication de certificat	40
Utilisation des services de registre par les applications	40
Service d'émission de mini-certificats par défaut	41
Interface de sécurité	41
Configuration et personnalisation	42
Règles	42
Styles de connexion	43
Connexion d'égal à égal.	44
Connexion client-serveur	44
Styles de connexion multiple	45
Classes	45
Chargement des applications	45
Chapitre 6. Réseaux MQSeries Everyplace et MQSeries	47
Interface avec MQSeries	47
Conversion de message.	54
Fonction	55
Compatibilité	55
Livraison garantie unique	56
Chapitre 7. Applications et utilitaires.	57
Postcard	57
MQSeries Everyplace Explorer	57
Chapitre 8. Interfaces de programmation	59
Annexe. Remarques	61
Marques	62
Glossaire	63
Bibliographie	67
Index	69

Figures

1.	La famille MQSeries	5
2.	Configurations hôte simple et distribuées.	6
3.	Configurations de station de travail classiques	7
4.	Configurations d'unités classiques	8
5.	Réseaux MQSeries Everyplace simples.	32
6.	Réseau MQSeries Everyplace en étoile.	33
7.	Hierarchie des objets pont MQSeries	48

Tableaux

1.	Environnements logiciels pris en charge par la version 1	3
2.	Éléments MQSeries Everyplace et MQSeries	9
3.	Propriétés des objets zone et de leurs éléments constitutifs	15
4.	Propriétés de l'objet attribut	15
5.	Propriétés de l'objet message	16
6.	Zones d'objet message avec provision	16
7.	Propriétés des files d'attente	21
8.	Propriétés d'un gestionnaire de files d'attente local	24
9.	Propriétés de connexion (gestionnaire de files d'attente éloigné)	24
10.	Opérations de messagerie sur les file d'attente MQSeries Everyplace	25
11.	Classes de message d'administration	26
12.	Structure générique d'un message administratif	27
13.	Support de l'authentification, du chiffrement et de la compression	35
14.	Styles de connexion	43
15.	Options de classes	45
16.	Propriétés de l'objet pont	48
17.	Propriétés du pont	49
18.	Propriétés du proxy du gestionnaire de files d'attente MQSeries	49
19.	Propriétés du service de connexion client	50
20.	Propriétés du programme d'écoute	51
21.	Propriétés des files d'attente éloignées MQSeries	52

A propos de ce manuel

Ce manuel est une présentation générale de MQSeries Everyplace. Il traite des concepts du produit et de ses relations avec d'autres produits MQSeries.

Pour plus d'informations sur l'API MQSeries Everyplace et son utilisation pour créer des applications MQSeries Everyplace, reportez-vous aux manuels MQSeries Everyplace Programming Reference et MQSeries Everyplace Programming Guide.

Lecteurs concernés

Ce manuel est destiné à quiconque s'intéresse à l'utilisation d'applications de messagerie sécurisées sur des unités légères, telles que capteurs, téléphones, assistants numériques personnels (PDA) et ordinateurs portables.

Connaissances requises

Aucune connaissance préalable n'est requise, mais une compréhension des concepts de messagerie sécurisée constitue un atout.

Si vous ignorez tout de ces concepts, nous vous conseillons de consulter les manuels MQSeries suivants :

- Introduction à MQSeries
- MQSeries pour Windows NT - Mise en route - Version 5.1

Ces manuels sont disponibles sous forme électronique dans la section Manuels de la bibliothèque MQSeries en ligne. Cette bibliothèque est accessible via le site Web de MQSeries, à l'adresse <http://www.ibm.com/software/ts/MQSeries/library/>

Terminologie

Voici la définition de quelques termes utilisés tout au long de ce manuel :

Famille MQSeries

désigne la collection de produits MQSeries décrits au «Chapitre 3. La famille MQSeries» à la page 5.

MQSeries Messaging

désigne le groupe des quatre produits de messagerie MQSeries décrits auxref `refid="mqfam">`.

MQSeries

désigne les trois groupes de produits de messagerie MQSeries suivants :

- Messagerie distribuée
- Messagerie hôte
- Messagerie de la station de travail

MQSeries Everyplace

désigne le quatrième groupe de produits de messagerie MQSeries, la messagerie diffuse.

Unité Ordinateur de taille quelconque exécutant des programmes MQSeries Everyplace, mais dépourvu d'objet **gestionnaire de canal MQSeries Everyplace** ou **programme d'écoute de canal MQSeries Everyplace** installé.

Remarque : Au niveau de l'octroi de licence, unité est synonyme de client Everyplace MQSeries.

Passerelle

Ordinateur de taille quelconque exécutant des programmes MQSeries Everyplace, et doté d'un objet **gestionnaire de canal MQSeries Everyplace** ou **programme d'écoute de canal MQSeries Everyplace** installé.

Remarque : Au niveau de l'octroi de licence, passerelle est synonyme de serveur MQSeries Everyplace.

Chapitre 1. Généralités

MQSeries Everyplace fait partie de la famille MQSeries des produits de messagerie professionnelle. Il est conçu pour répondre non seulement aux besoins en messagerie des unités légères, telles que les capteurs, téléphones, assistants numériques personnels (PDA) et ordinateurs portables, mais également aux demandes de connexion mobile et aux conditions induites par l'utilisation de réseaux de communication fragiles. Il assure la livraison unique garantie standard de MQSeries et l'échange de messages avec les autres membres de la famille. Dans la mesure où nombre d'applications MQSeries Everyplace sont exécutées sans la protection d'un pare-feu Internet, il propose également des fonctions de sécurisation sophistiquées.

Les unités légères requièrent un système de messagerie peu gourmand en ressources système, et MQSeries Everyplace est optimisé au niveau de l'empreinte système et de l'efficacité du protocole. Il n'offre pas les mêmes fonctions aux autres membres de la messagerie de la famille MQSeries, mais assure la transparence des opérations avec eux. MQSeries Everyplace est pourvu de toutes les extensions possibles en ce qui concerne la mobilité, les déplacements, l'accès aux messages en local et à distance, la sécurité et le support de la messagerie sur des réseaux non fiables.

MQSeries Everyplace fait partie de la famille d'informatique diffuse IBM et se trouve de ce fait parfaitement adapté à l'intégration avec les autres produits diffus et sans fil d'IBM.

Remarques sur la version 1.0

- MQSeries Everyplace version 1.0 est un ensemble d'outils de développement qui permet aux utilisateurs d'écrire des applications MQSeries Everyplace et de créer un environnement dans lequel les exécuter.
- Dans cette version, le déploiement de MQSeries Everyplace sur des unités diffuses incombe au fournisseur de l'application et de la solution.

Chapitre 2. Conditions préalables

Le tableau 1 indique les environnements logiciels utilisables pour exécuter MQSeries Everyplace version 1.¹

Tableau 1. Environnements logiciels pris en charge par la version 1

	Système d'exploitation
Unité	EPOC
	Palm OS
	Windows CE
	Windows 95 Windows 98 Windows NT v4 Windows 2000
Passerelle	Windows NT v4 Windows 2000
Remarques :	
1. La version 1.0 est fournie dans Java pour toutes les plateformes qui prennent en charge Java.	
2. Une fonction client limitée, qui ne fournit que l'accès synchrone aux files d'attente éloignées, est disponible comme base de code C, sur Palm OS uniquement.	
3. Java 1.1, au niveau le plus récent disponible pour la plateforme, est conseillé. Pour plus d'informations sur les niveaux de Java testés, consultez le site Web de MQSeries Everyplace (www.ibm.com/software/mqseries/everyplace).	

1. Le code d'unité MQSeries Everyplace peut être exécuté sur toute unité qui exécute Java, mais il n'a été testé que sur les systèmes d'exploitation répertoriés dans le tableau 1.

Chapitre 3. La famille MQSeries

La famille MQSeries comporte de nombreux produits, offrant une vaste gamme de fonctions, comme illustré à la figure 1

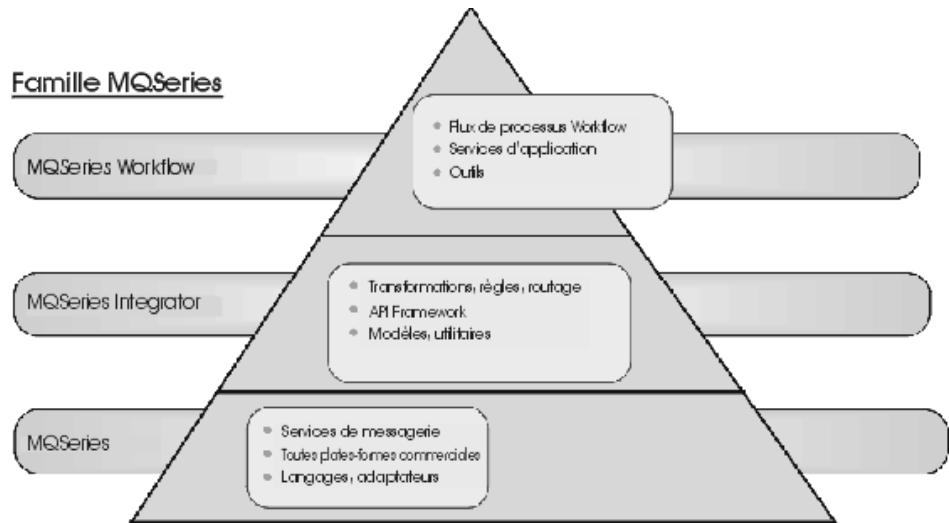


Figure 1. La famille MQSeries

- **MQSeries Workflow** simplifie l'intégration à l'échelle de l'entreprise en automatisant les processus métier qui impliquent personnes et applications.
- **MQSeries Integrator** est un puissant logiciel de routage de messages, qui assure l'acheminement intelligent, basé sur des règles, en temps réel, des messages, ainsi que la transformation et le formatage des contenus.
- **MQSeries Messaging** fournit une connectivité indifférenciée, du bureau à l'ordinateur principal, grâce à une messagerie professionnelle, avec plus de 35 plateformes prises en charge.

Les produits MQSeries Workflow et MQSeries Integrator bénéficient de la connectivité fournie par la couche de messagerie de MQSeries.

La messagerie de la famille MQSeries est fournie par les produits MQSeries (MQS) et MQSeries Everyplace ; chacun étant conçu pour prendre en charge une ou plusieurs plateformes serveur matérielles et/ou systèmes d'exploitation associés. Du fait de la variété des fonctionnalités des plateformes, ces produits individuels sont structurés en groupes de produit, par fonction et conception. Il existe quatre groupes de produits de ce type :

- **Messagerie distribuée** : MQSeries pour Windows NT, AIX, AS/400, HP-UX, Sun Solaris et autres plateformes
- **Messagerie hôte** : MQSeries pour OS/390

- **Messagerie de station de travail** : MQSeries pour Windows 3.1, 95, 98
- **Messagerie diffuse** : MQSeries Everyplace

La messagerie en elle-même, indépendamment du produit ou du groupe de produits spécifique, est basée sur les gestionnaires de files d'attente. Ces gestionnaires gèrent des files d'attente qui peuvent contenir des messages. Les applications communiquent avec un gestionnaire de files d'attente locale, et récupèrent ou placent des messages dans les files d'attente. Si un message est placé dans une file d'attente éloignée, appartenant à un gestionnaire de files d'attente éloigné, le message est transmis à ce gestionnaire via des canaux. Ainsi les messages peuvent passer par un ou plusieurs gestionnaires intermédiaires avant d'atteindre leur destinataire. Le principe de base de la messagerie est de "découpler" l'application expéditrice de l'application destinataire, en mettant les messages en file d'attente en route si nécessaire. Tous les produits de messagerie MQSeries reposent sur les mêmes éléments fondamentaux (gestionnaire de files d'attente, files d'attente, messages et canaux), malgré des différences notables dans les détails.

L'hôte MQSeries et les produits de messagerie distribuée prennent en charge de nombreuses configurations réseau, qui impliquent toutes des clients et des serveurs², dont vous trouverez quelques exemples à la figure 2.

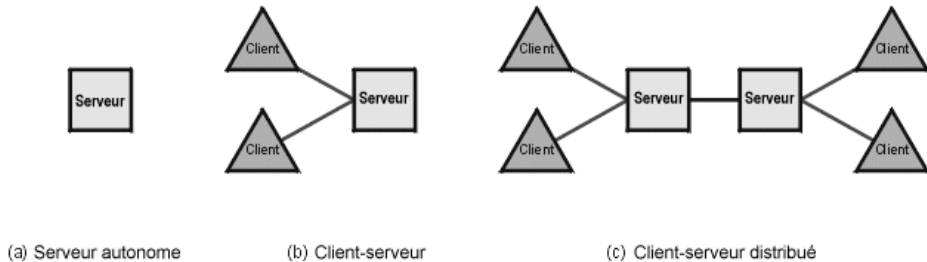


Figure 2. Configurations hôte simple et distribuées

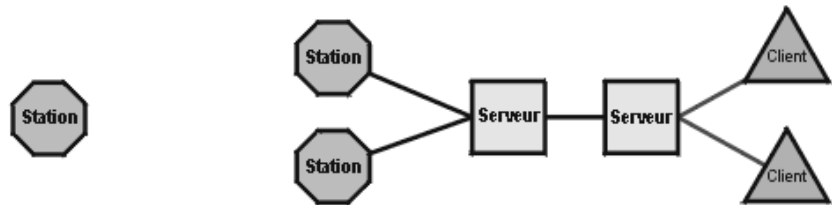
Dans le cas le plus simple, un serveur autonome exécutant un gestionnaire de files d'attente est configuré. Une ou plusieurs applications, exécutées sur ce serveur, échangent des messages via des files d'attente. La configuration client-serveur est une autre configuration possible. Dans ce cas, le gestionnaire de files d'attente est localisé sur le serveur, mais est accessible aux clients via un canal client. Le canal client est une liaison bidirectionnelle sous un protocole MQSeries unique, implémentant un lien semblable à un appel de procédure éloignée (RPC). Les applications peuvent être exécutées sur les clients, avec accès aux files d'attente du serveur. Un des avantages de la configuration client-serveur est la légèreté de l'infrastructure de la messagerie du client, du fait qu'elle s'appuie sur le gestionnaire de files d'attente du serveur. L'inconvénient est que les clients et leur serveur associé opèrent de façon synchrone et requièrent de ce fait que le canal client soit disponible en permanence.

2. Notez que ces termes ont une signification spécifique au sein de MQSeries, qui ne correspond pas toujours à leur utilisation habituelle. Dans ce document, ils sont toujours utilisés dans leur acception pour MQSeries.

La configuration client-serveur distribuée est plus complexe, dans la mesure où plusieurs serveurs entrent en jeu. Dans ce type de configuration, l'échange de messages entre serveurs passe par des canaux de transmission de messages. Ces canaux de transmission sont unidirectionnels, avec un protocole conçu pour un échange sécurisé, asynchrone, des données des messages. Les clients peuvent poursuivre leurs traitements sans que ces canaux de transmission soient disponibles, mais aucun message ne peut alors transiter entre les serveurs.

Les produits de messagerie pour station de travail MQSeries proposent un sous-ensemble de ces options de configuration. Elles prennent en charge, à la place de serveurs, des stations de travail dotées d'un gestionnaire de files d'attente, mais ne permettent pas la connexion de clients. Elles peuvent toutefois être connectées à d'autres stations de travail, ou même à des serveurs, via les canaux de transmission de messages MQSeries. Les stations de travail sont de ce fait considérées comme des serveurs légers, utilisés à la place de clients lorsque la possibilité de travail en mode asynchrone est requise.

Deux configurations classiques de stations de travail sont illustrées à la figure 3. Dans le cas (b), les applications de la station de travail peuvent être exécutées indépendamment des serveurs et des clients :



(a) Station de travail autonome

(b) Client/Station de travail/Client distribué

Figure 3. Configurations de station de travail classiques

Le produit MQSeries Everyplace de messagerie diffuse prend en charge les configurations via la mise à disposition d'unités et de passerelles.

L'unité MQSeries Everyplace est un ordinateur exécutant le code MQSeries Everyplace sans gestionnaire de canal. Ce qui signifie qu'une unité ne peut communiquer qu'avec une seule autre unité ou passerelle à la fois. Les unités MQSeries Everyplace peuvent être de taille extrêmement réduite (capteur sur un pipeline, par exemple), moyenne (téléphone, assistant personnel (PDA) ou ordinateur portable) ou plus conséquente (ordinateur de bureau et stations de travail). Ces unités sont souvent appelées unités diffuses, bien que ce terme suppose des limites de taille et de capacité non applicables à ce produit.

Une passerelle est un ordinateur qui exécute le code MQSeries Everyplace avec un gestionnaire de canal MQSeries Everyplace ou un programme d'écoute de canal

MQSeries Everyplace configuré. Une passerelle offre, outre toutes les fonctions induites par le code de l'unité, la possibilité de communiquer simultanément avec plusieurs passerelles d'unité. Les passerelles fournissent également le mécanisme d'échange de messages entre un réseau MQSeries Everyplace et un réseau MQSeries.

Pour simplifier, on peut dire que les unités combinent de nombreux attributs des clients et des serveurs. Il est possible de les configurer avec des fonctions complètes de mise en file d'attente, les rendant ainsi capables de fonctionner en mode asynchrone. Elles peuvent également accéder à des files d'attente éloignées, une fonction qui présente quelques similitudes avec l'accès client aux files d'attente de serveur. Contrairement aux serveurs, les unités n'assurent pas de connexions avec les clients. Les unités communiquent directement entre elles via une fonction de messagerie d'égal à égal. Elles communiquent également via des canaux de transmission, mais qui sont propres à MQSeries Everyplace et appelés canaux dynamiques pour les distinguer des canaux client MQSeries et des canaux de transmission de messages MQSeries. Les canaux dynamiques sont bidirectionnels et prennent en charge l'ensemble des fonctions fournies par MQSeries Everyplace, messagerie synchrone et asynchrone comprises.

Les passerelles prennent obligatoirement en charge les canaux dynamiques MQSeries Everyplace pour communiquer avec les unités. Elles peuvent en outre prendre en charge les canaux client MQSeries pour communiquer avec les serveurs. Comme les serveurs, les passerelles sont dotées de gestionnaire de files d'attente et, de ce fait, peuvent prendre en charge les applications de messagerie locales.

Quelques configurations diffuses classiques sont illustrées à la figure 4

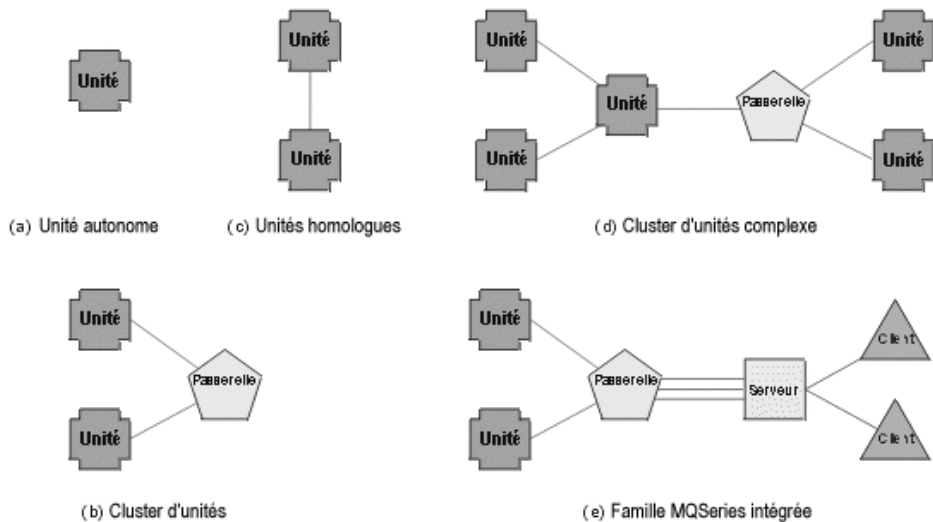


Figure 4. Configurations d'unités classiques

Dans le cas de la figure 4 (b), une passerelle sert à interconnecter les unités. Les passerelles présentent l'avantage de pouvoir gérer simultanément plusieurs demandes

de connexion entrantes, contrairement aux unités qui ne peuvent en gérer qu'une à la fois. Passerelles et unités peuvent toutes deux émettre simultanément plusieurs demandes de connexion sortantes. Si, dans la configuration (b), une unité avait été utilisée à la place de la passerelle, les deux unités terminales auraient dû attendre leur tour pour contacter cette unité intermédiaire, même si elle les avait contactées simultanément. Dans le cas (d), une unité et une passerelle sont toutes deux utilisées pour relier les unités. Dans le cas (e), une passerelle sert à relier un réseau d'unités à un serveur MQSeries, une configuration dans laquelle les messages peuvent naviguer entre tous les composants, unités, passerelles, serveurs, stations de travail et clients. Les caractéristiques essentielles de ces composants sont recensées dans le tableau 2.

Tableau 2. Eléments MQSeries Everyplace et MQSeries

Composant	Caractéristiques	Proposé par
Unité (MQSeries Everyplace)	<p>Fournit une fonction de messagerie garantie pour les applications via des canaux dynamiques</p> <ul style="list-style-type: none"> autorise l'accès synchrone aux files d'attente locales et éloignées autorise la livraison asynchrone aux files d'attente éloignées limitée à la gestion d'une demande entrante à la fois 	Diffuse
Client (MQS)	<p>Fournit une fonction de messagerie garantie pour les applications locales</p> <ul style="list-style-type: none"> requiert une connexion à un serveur par canal client synchrone autorise l'accès synchrone aux files d'attente sur le serveur connecté exclusivement autorise la livraison asynchrone aux files d'attente éloignées via le serveur connecté 	Hôte distribué
Passerelle (MQSeries Everyplace)	<p>Fournit une fonction de messagerie garantie pour les applications via des canaux dynamiques</p> <ul style="list-style-type: none"> autorise l'accès synchrone aux files d'attente locales et éloignées autorise la livraison asynchrone aux files d'attente éloignées peut gérer simultanément plusieurs demandes entrantes <p>Prend en charge la connexion de plusieurs serveurs MQSeries via les canaux client</p>	Diffuse

Tableau 2. Eléments MQSeries Everyplace et MQSeries (suite)

Composant	Caractéristiques	Proposé par
Serveur (MQS)	<p>Fournit une fonction de messagerie garantie pour les applications via des canaux de transmission de messages</p> <ul style="list-style-type: none"> autorise l'accès synchrone aux files d'attente locales autorise la livraison asynchrone aux files d'attente éloignées <p>Prend en charge la connexion de plusieurs clients MQSeries via les canaux client</p>	Hôte distribué
Station de travail (MQS)	<p>Fournit une fonction de messagerie garantie pour les applications via des canaux de transmission de messages</p> <ul style="list-style-type: none"> autorise l'accès synchrone aux files d'attente locales autorise la livraison asynchrone aux files d'attente éloignées 	Station de travail

Chapitre 4. Conditions requises

Ce chapitre décrit les conditions qui ont régi la conception et la mise en oeuvre de MQSeries Everyplace.

Capacités

MQSeries Everyplace étend la portée de la messagerie de la famille MQSeries :

- En prenant en charge les unités d'entrée de gamme, telles qu'assistants numériques personnels (PDA), téléphones ou capteurs, les autorisant à intégrer le réseau de messagerie MQSeries. Il prend également en charge des unités intermédiaires (ordinateurs portables, stations de travail et certaines plateformes distribuées). MQSeries Everyplace offre la même qualité de service, la livraison garantie unique, et permet l'échange de messages avec d'autres membres de la famille.
- En fournissant des fonctions de sécurisation étendues pour protéger les messages, les files d'attente et les données, pendant leur stockage ou leur transmission.
- En fonctionnant efficacement dans des environnements de communication hostiles, avec des réseaux instables ou une bande passante soumise à de fortes contraintes. Il est doté d'un protocole de communication par fil efficace et assure la reprise automatique en cas d'échec de la liaison.
- En prenant en charge les utilisateurs nomades, avec transfert des points de connectivité du réseau au fur et à mesure que les unités sont déplacées. Il contrôle également le fonctionnement dans des conditions inhabituelles : limitation ou défaillance des ressources de la batterie ou des réseaux.
- En opérant au travers de pare-feu configurés en conséquence.
- En réduisant au minimum les tâches d'administration de l'utilisateur, de sorte que la présence de MQSeries Everyplace sur une unité soit quasiment transparente. MQSeries Everyplace constitue ainsi une base adaptée à la construction d'applications de type utilitaire.
- Par la simplicité de sa personnalisation et son extension, via l'utilisation de règles fournies par l'application et d'autres classes qui modifient le comportement, ou via la transformation en sous-classes de classes d'objets de la base, par exemple, pour représenter différents types de messages.

Applications

Il n'existe pas de liste restrictive des applications MQSeries Everyplace possibles, le choix étant vaste et varié, mais un nombre non négligeable peut être considéré comme des applications personnalisées développées pour des groupes d'utilisateurs spécifiques. La liste suivante donne un aperçu significatif :

- **Applications consommateur** : shopping en ligne via un PDA, recherche d'informations sur les vols des différents compagnies aériennes, transactions financières à partir d'un téléphone portable.
- **Applications de contrôle** : collecte et intégration de données, à partir des capteurs sur les pipelines pétroliers, transmises par satellite, télécommande d'équipement (valves, par exemple) sécurisée garantissant la validité de l'intervenant.

- **Travail mobile** : visite de professionnels (agent d'assurance), émission rapide d'un reçu client pour les entreprises de transport de colis, serveur de restauration rapide échangeant des informations avec la cuisine, suivi des scores d'un tournoi de golf, système de messagerie mobile sécurisé pour la police, informations professionnelles pour les ouvriers d'un chantier où la communication est souvent coupée, lecture des compteurs domestiques.
- **Productivité personnelle** : duplication de courrier/agenda, réplication de bases de données, décentralisation du traitement sur ordinateur portable.

Exigences des clients

La conception de MQSeries Everyplace a intégré les exigences des clients, au rang desquels :

- **Administration** : configuration et maintenance minimales ; prise en charge de l'administration locale et à distance ; possibilité d'extension et de personnalisation des fonctions administratives pour répondre aux besoins d'applications spécifiques ; amélioration de la reconnaissance et de la reprise automatiques ; disponibilité d'éléments d'administration indépendants, utilisables au choix.
- **Communications** : protocole par fil très efficace ; en-tête minimaux ; pas de zones obligatoires dans les messages (à l'exception d'un identificateur unique) ; possibilité de changer le codage des données ; support pour la compression, le chiffrement et l'authentification ; négociation de bout en bout des caractéristiques de compression et de sécurité ; possibilité de passer outre les pare-feu ; adaptateurs de communications enfichables.
- **Compatibilité** : qualité du service et fluidité des échanges de messages ; possibilité de communiquer avec d'autres systèmes MQSeries sans modification de l'application ; contrôle souple de l'échange de messages entre MQSeries et MQSeries Everyplace.
- **Empreinte** : nettement moins de 100 ko pour la configuration minimale.
- **Fonction** : transfert de messages en mode synchrone et asynchrone, accès aux messages dans les files d'attente locales et éloignées ; possibilité d'utiliser n'importe quelle zone du message pour la recherche sélective ; contrôle sélectif du support de sauvegarde d'une file d'attente.
- **Support de règle** : contrôle de nombreuses facettes du comportement par le biais de règles : moment d'envoi des messages, nombre maximal de relances d'une communication, action à effectuer lorsqu'un message est trop long ou que la file d'attente cible est saturée.
- **Sécurité** : support intégral de la sécurité, de l'authentification et de l'irréfutableté ; sécurité au niveau message et au niveau file d'attente ; protection du système de messagerie contre les attaques ; sécurité enfichable utilisant les algorithmes standard de l'industrie ; possibilité d'intégration avec les accréditations utilisateur du système d'exploitation ; possibilité de mise en conformité avec les normes de sécurité nationales, avec changement possible du support de sécurité lorsque les messages passent les frontières.

Chapitre 5. Concepts du produit

Introduction

Les éléments constitutifs du modèle de programmation de MQSeries Everyplace sont les messages, les files d'attente et les gestionnaires de files d'attente. Les messages MQSeries Everyplace sont des objets dont le contenu est défini par l'application. Stockés, ils sont maintenus en file d'attente et peuvent être déplacés au sein d'un réseau MQSeries Everyplace. Pour envoyer des messages vers une file d'attente cible, il suffit de préciser le gestionnaire des files d'attente cible et le nom de la file d'attente. Les applications placent les messages en files d'attente par le biais d'une opération "put", leur récupération étant généralement effectuée par une opération "get". Les files d'attente, locales ou éloignées, sont gérées par des gestionnaires de files d'attente. Les unités comme les passerelles stockent les données de configuration dans un registre.

Les applications sur les unités peuvent exploiter une ou toutes les API et fonctions disponibles sur l'unité : elles ne sont pas limitées aux interfaces de programmation MQSeries Everyplace. Grâce aux canaux dynamiques, les unités MQSeries Everyplace peuvent se connecter à d'autres unités MQSeries Everyplace et/ou à une passerelle MQSeries Everyplace.

Les applications sur les passerelles peuvent également exploiter une ou toutes les API et fonctions disponibles sur la passerelle : elles ne sont pas limitées aux interfaces de programmation MQSeries Everyplace. Grâce aux canaux dynamiques, une passerelle peut être connectée aux autres passerelles et/ou aux unités MQSeries Everyplace. Par les canaux client MQSeries, une passerelle peut être connectée à un ou plusieurs serveurs MQSeries (mais non aux autres passerelles MQSeries Everyplace). MQSeries Everyplace et MQSeries peuvent coexister sur une seule machine, la présence de MQSeries étant néanmoins totalement facultative.

Les capacités des unités et des passerelles à fonctions intégrales sont identiques, à ceci près que :

- Les passerelles peuvent gérer plusieurs demandes entrantes simultanées (issues d'autres unités et/ou passerelles).
- Les passerelles peuvent simultanément assurer l'interface avec plusieurs serveurs MQSeries.

Les canaux dynamiques prennent en charge les connexions réseau suivantes :

- Connexions à accès commuté entrantes.
- Connexions permanentes, telles qu'un LAN classique, une ligne spécialisée, un LAN infrarouge ou sans fil.

Les protocoles de communication sont mis en oeuvre par un jeu d'adaptateurs, un par protocole pris en charge. Il est ainsi possible d'ajouter de nouveaux protocoles en fonction des besoins et d'adapter l'empreinte mémoire à une configuration spécifique dans un environnement donné.

Les files d'attente sont mappées individuellement aux supports de stockage via un autre jeu d'adaptateurs. Une file d'attente peut ainsi être stockée dans le système de fichiers ou dans la mémoire, selon l'adaptateur choisi.

L'interface de programmation MQSeries Everyplace est conçue de sorte que les applications puissent être écrites indépendamment de l'emplacement des files d'attente. Un programme conçu pour accéder aux files d'attente locales peut être exécuté tel quel à partir d'un gestionnaire de files d'attente éloigné (sous réserve de respect des conditions de sécurité et du fait que certaines opérations MQSeries Everyplace ne sont pas prises en charge sur les files d'attente éloignées). Cette indépendance concerne l'exécution de toutes les fonctions administratives.

Objets message

Les objets message MQSeries Everyplace sont fondamentalement différents des messages pris en charge par MQSeries. Dans MQSeries, les messages sont des tableaux d'octets, divisés en en-tête de message et corps de message. L'en-tête de message, interprété par MQSeries, contient des informations essentielles, telles que l'identité de la réponse à la file d'attente, la réponse au gestionnaire de files d'attente, l'ID du message et l'ID de corrélation ; le corps du message n'est pas interprété.

A l'inverse, les messages dans MQSeries Everyplace sont des objets message hérités d'un objet MQSeries Everyplace appelé objet zone. Les messages sont de véritables objets, sans le concept d'en-tête ou de corps de message. Pour expliciter la nature réelle de l'objet message, il convient de bien comprendre le concept de zones de base. Ces objets zone, abondamment utilisés dans MQSeries Everyplace, sont un cumul de zones, chaque zone comprenant un nom, un type de données et les données elles-mêmes. Les noms de zones sont des chaînes de caractères ASCII (à l'exception de quelques caractères réservés) de longueur illimitée.

Les types de zones possibles sont les suivants :

- **ASCII** chaîne ou tableau dynamique de chaînes ASCII
- **Boolean** valeur booléenne
- **Byte**, tableau fixe ou dynamique d'octets
- **Double floating point**, tableau fixe ou dynamique de valeurs décimales double à virgule flottante
- **Fields** objets zones ou tableau dynamique d'objets zone (l'imbrication d'objets zone est admise)
- **Floating point** virgule flottante ou tableau fixe ou dynamique de valeurs de type virgule flottante
- **Integer** entier (4 octets), ou tableau fixe ou dynamique d'entiers
- **Long integer** entier long (8 octets), ou tableau fixe ou dynamique d'entiers longs
- **Short integer** entier court (2 octets), ou tableau fixe ou dynamique d'entiers courts
- **UNICODE** chaîne UNICODE ou tableau dynamique de chaînes UNICODE

Un type est associé aux objets zone, qui correspond au nom de la classe de l'objet de programmation. Les descendants de cette classe d'objets sont utilisés les programmes

d'application comme objets message et MQSeries Everyplace utilise leur type pour instancier la classe d'objet adéquate lorsque nécessaire, par exemple après transmission d'un objet message via un canal.

Les objets zone fournissent plusieurs méthodes. Ainsi, les zones peuvent être énumérées ou leur existence vérifiée. De même, l'égalité de deux zones peut être testée. Les objets zone ont la capacité de supprimer et de restaurer les éléments de leur zone de et vers un tableau d'octets, utilisé par exemple pour fournir les données requises pour transférer l'objet via une liaison et le restaurer après transmission. Les méthodes de suppression et de restauration peuvent être remplacées de sorte que les objets zone effectuent leur sérialisation par d'autres moyens, pour, par exemple, interroger une base de données sur le contenu de leur zone au moment de la transmission. Le tableau 3 recense les propriétés des objet zone et/ou de leurs zones constitutives.

Tableau 3. Propriétés des objets zone et de leurs éléments constitutifs

Propriété	Présence	
	Objets zone	Zones
Objet attribut associé	facultatif	
Zone(s) constitutive(s)	oui	
Caché		oui
Nom		oui
Type	oui	oui
Valeur		oui

La propriété "caché" permet d'ignorer une zone dans une opération de comparaison.

Les objets attribut contiennent les mécanismes d'exécution des processus d'authentification, de chiffrement et de compression, et peuvent être associés à des objets zone.

- **Authentification** : contrôle l'accès
- **Compression** : réduit les besoins mémoire (pour la transmission et/le stockage)
- **Chiffrement** : protège le contenu lorsque l'objet est supprimé (et autorise sa restauration)

Les objets attribut, au coeur du modèle de sécurité MQSeries Everyplace, autorisent notamment l'accès sélectif au contenu et la protection des données de sauvegarde. Le tableau 4, recense les propriétés des objets attribut. La valeur Rule, le cas échéant, contrôle les opérations autorisées.

Tableau 4. Propriétés de l'objet attribut

Propriété	Présence
Cryptor	facultatif (obligatoire dans certains cas)
Authenticator	facultatif
Compressor	facultatif

Tableau 4. Propriétés de l'objet attribut (suite)

Propriété	Présence
Rule	facultatif
Type	facultatif

Les objets message sont dérivés des objets zone et comportent un UID (identificateur unique) généré par MQSeries Everyplace. Cet UID, qui identifie de manière unique un objet message, est constitué :

- **du nom** du gestionnaire de files d'attente émetteur (ajouté par le gestionnaire de files d'attente à réception de l'objet). Ce nom doit être globalement unique.
- **de l'heure** de création de l'objet message (ajouté à la création).

Les propriétés de base des objets message sont recensées dans tableau 5, sans compter celles dont ils héritent comme objets zone.

Tableau 5. Propriétés de l'objet message

Propriété	Explication
Msg_OriginQMgr	Nom du gestionnaire de files d'attente qui a envoyé le message
Msg_Time	Heure de création de l'objet message par l'application

Ces deux propriétés constituent l'identificateur unique (UID) de l'objet message.

Aucune autre information n'est requise dans un message destiné à un autre gestionnaire de files d'attente MQSeries Everyplace, bien que d'autres zones doivent généralement être intégrées pour transmettre les informations des messages. Le plus souvent, les messages sont des descendants de la classe de l'objet message de base et à ce titre, comportent des zones supplémentaires adaptées. Nombre de ces zones supplémentaires sont communes à une grande partie des applications, par exemple "réponse au gestionnaire de files d'attente". De même, MQSeries Everyplace assure en partie leur prise en charge.

Le tableau 6 répertorie les zones prises en charge.

Tableau 6. Zones d'objet message avec provision

Nom de la zone	Utilisation
Msg_CorrelID	Chaîne de caractères binaires généralement utilisée pour associer une réponse au message d'origine.
Msg_ExpireTime	Délai au bout duquel le message peut être supprimé (même s'il n'a pas été livré).
Msg_LockID	Clé requise pour déverrouiller un message.
Msg_MsgID	Utilisé par l'application pour assurer l'association avec le message d'origine.
Msg_Priority	Priorité du message.

Tableau 6. Zones d'objet message avec provision (suite)

Nom de la zone	Utilisation
Msg_ReplyToQ	Nom de la file d'attente dans laquelle la réponse au message doit être envoyée.
Msg_ReplyToQMgr	Nom du gestionnaire de files d'attente auquel la réponse au message doit être envoyée.
Msg_Resend	Indique que le message est une réexpédition d'un message précédent.
Msg_Style	Distingue les commandes des demandes/réponses, etc.

Dans tous les cas, une constante définie permet de transférer le nom de la zone dans un seul octet. Certaines zones bénéficient de dispositions étendues - par exemple : la priorité (le cas échéant) joue sur l'ordre de transmission des messages : l'ID de corrélation déclenche l'indexation d'une file d'attente sur les valeurs de cette zone pour accélérer les recherches ; le délai d'expiration déclenche la péremption du message, etc.

Plusieurs méthodes sont définies pour les objets message, par exemple la capacité d'extraire l'UID du message, le nom du gestionnaire de files d'attente émetteur et l'heure de création de l'objet. D'autres méthodes utiles sont héritées de la classe de l'objet zone, par exemple, diverses méthodes pour récupérer et définir les valeurs des zones. Particulièrement intéressante, la méthode dump permet de vider les données de l'objet dans une chaîne de caractères binaires. MQSeries Everyplace appelle cette méthode lorsqu'un message doit être sauvegardé en mémoire persistante ou qu'il doit être transmis via un canal dynamique. Ainsi c'est l'objet message qui détermine lui-même la représentation externe de la valeur de ses données, ce qui est exploitable de diverses manières. Par exemple, un objet peut simplement vider les valeurs de ses zones constitutives ou peut décider d'interroger plutôt une base de données. La méthode complémentaire restore offre des fonctions de contrôle similaires lorsqu'un objet est recréé à partir de son format de vidage. Notez que si l'objet message est affecté d'un objet attribut, les modules de codage, de chiffrement et de compression de donnée de l'attribut sont appelés au moment du vidage ; de même, les modules de décodage, de déchiffrement et de décompression sont appelés au moment de la restauration.

Lorsque MQSeries Everyplace transmet un objet message, pour réduire la charge sur le câble, il ne transmet pas le fichier de classe associé. De la même façon, la classe de message appropriée doit être disponible sur chacun des gestionnaires de files d'attente sur lesquels l'objet message va être instancié.

La méthode de vidage d'objet message par défaut a été optimisée pour minimiser la taille de la chaîne de caractères binaires générée, afin d'optimiser le stockage et la transmission de message.

Format des données de vidage

Le format des données de vidage par défaut code les zones comme suit :

```
{Length Identifler Fence {Data}} {Length Identifler Fence {Data}} { ...}
```

où :

- Data : valeur des données. Les entiers sont compressés par suppression des 0 et des F de début. Les valeurs booléennes ne sont pas associées à des octets de données.
- Fence : octet spécial délimitant la frontière entre l'identificateur et l'élément Data facultatif. Cet octet indique également le type de l'élément Data.
- Identifler : conserve le nom de zone dans une chaîne ASCII de longueur variable, terminée par un octet de fin.
- Length : indique la longueur de la zone de données, qui peut varier de 1 à 4. Les deux premiers bits du premier octet sont réservés à la longueur de la zone Length. Longueurs admises : entre 0 et 1 073 741 823.

Le résultat est un flot de données extrêmement compact. La compression des données permet d'économiser encore de la place. La compression XOR avec un flot d'octets antérieur peut donner de bons résultats, mais du fait de nature variable de ces zones et de l'ordre également variable des zones, un XOR simple peut ne pas produire les résultats escomptés. MQSeries Everyplace intègre un XOR intelligent, fonctionnant zone par zone, ce qui ne peut qu'améliorer la compression.

Files d'attente

Les files d'attente sont essentiellement utilisées pour contenir les objets message en attendant que les programmes d'application les récupèrent. Comme les messages, les files d'attente dérivent d'objets zone. L'accès direct à l'objet file d'attente par les applications n'est pas admis³ ; le gestionnaire de files d'attente joue le rôle d'intermédiaire entre les programmes d'application et les files d'attente. Les files d'attente sont identifiées par leur nom, chaîne de caractères ASCII de taille non limitée⁴, mais qui doit être unique au sein d'un gestionnaire de files d'attente spécifique.

MQSeries Everyplace prend en charge de nombreux types de file d'attente :

Files d'attente locales

Les files d'attente locales sont utilisées par les applications pour stocker les messages en toute sécurité. Elles offrent un espace de stockage accessible via une classe adaptateur, généralement la classe adaptateur du disque. MQSeries Everyplace fournit en outre une classe adaptateur de mémoire qui maintient l'espace de stockage en mémoire, pour des accès rapides (avec le risque de perte des messages en cas de panne du système). En créant l'adaptateur approprié, vous pouvez stocker les messages n'importe où, sur une base file d'attente par file d'attente, par exemple dans une base de

3. L'accès direct au sein d'une règle de file d'attente est autorisé.

4. Pour une meilleure interopérabilité il est conseillé toutefois de respecter les conventions de dénomination MQSeries, et se limiter les noms à 48 caractères. Le système de fichiers utilisé peut également avoir une incidence sur la longueur.

données relationnelle, sur un CD inscriptible, etc. Les files d'attente locales peuvent être exploitées en ligne ou hors ligne, c'est-à-dire connectées ou non à un réseau. Les accès et la sécurité sont propriété de la file d'attente et peuvent être octroyés par un gestionnaire de files d'attente éloigné (si vous êtes connecté à un réseau), permettant aux autres d'envoyer des messages vers la file d'attente ou d'en recevoir. L'accès à une file d'attente locale est toujours synchrone.

Files d'attente éloignées

Les files d'attente éloignées ne résident pas dans l'environnement local, mais il existe localement une définition qui identifie le gestionnaire de files d'attente propriétaire et la file d'attente réelle. L'accès aux files d'attente éloignées peut être synchrone ou asynchrone. S'il existe localement une définition de la file d'attente éloignée, le mode d'accès est basé sur cette définition. Sinon, une reconnaissance de file d'attente a lieu, de sorte que les caractéristiques soient découvertes et le mode d'accès forcé à synchrone.

Les files d'attente synchrones ne sont accessibles que lorsqu'elles sont connectées à un réseau doté d'un chemin vers le gestionnaire propriétaire de la file d'attente. Si la connexion au réseau n'est pas établie, les opérations telles que put, get et browse (reportez-vous au tableau 10 à la page 25) provoquent une erreur. La file d'attente propriétaire contrôle les droits d'accès et les impératifs de sécurité requis pour accéder à la file d'attente. C'est à l'application que revient la gestion des erreurs ou des tentatives d'envoi et de réception des messages. MQSeries Everyplace n'assure alors plus la livraison garantie unique.

Les files d'attente asynchrones peuvent recevoir et conserver des messages, sans qu'ils soient récupérés. Si la connexion réseau est établie, les messages sont alors envoyés au gestionnaire de files d'attente propriétaire et à la file d'attente. Sinon, les messages sont stockés localement jusqu'à établissement de la connexion, puis transmis. Les applications peuvent ainsi opérer sur la file d'attente lorsque l'unité est hors ligne, sous réserve que la file d'attente soit dotée d'un espace de stockage temporaire pour les messages.

Files d'attente de stockage et de réacheminement

Ce type de file d'attente stocke les messages jusqu'à ce qu'ils puissent être réacheminés au gestionnaire de files d'attente suivant (pas nécessairement le propriétaire). Ce type de file d'attente est généralement (mais pas forcément) défini sur une passerelle et l'unité doit récupérer ses messages lorsqu'elle se connecte au réseau. Une file d'attente de stockage et réacheminement peut conserver des messages destinés à plusieurs gestionnaires de files d'attente ou vous pouvez en définir une par gestionnaire de files d'attente. Lorsqu'un expéditeur souhaite envoyer un message à un destinataire qui est peut-être déconnecté, il l'envoie malgré tout au gestionnaire de files d'attente/file d'attente du destinataire ; le serveur intermédiaire détecte que le destinataire n'est pas connecté et conserve le message dans son espace de stockage local. Aucune modification de l'application expéditrice n'est requise pour envoyer un message à une file d'attente cible connectée ou déconnectée.

Files d'attente du serveur local

Ce type de file d'attente réside normalement sur une unité (supposée être connectée occasionnellement) et pointe sur une file d'attente de stockage et réacheminement sur un gestionnaire de files d'attente appelé serveur local. Cette file d'attente extrait les messages du serveur local dès que l'unité se connecte au réseau. Une fois le message extrait, elle le transmet au gestionnaire de files d'attente local via l'appel des méthodes `putMessage` et `confirmputMessage` (reportez-vous à la section «Opérations du gestionnaire de files d'attente» à la page 24). Il appartient alors au gestionnaire de files d'attente de placer le message dans la file d'attente locale appropriée. La méthode d'extraction des messages du serveur peut être nettement plus efficace en termes de flux sur le réseau que l'insertion des messages par le serveur ; en effet, la file d'attente du serveur local se sert de l'accusé de réception du premier message comme demande pour le message suivant (s'il y en a un), alors que la méthode d'insertion par le serveur requiert une demande/réponse pour envoyer le message et une seconde demande/réponse pour le flux de confirmation. Un intervalle d'interrogation est normalement défini pour les files d'attente du serveur local qui vérifie alors l'existence de messages en attente sur le serveur tant qu'il est connecté au réseau. L'intervalle d'interrogation est une option de configuration administrative.

Files d'attente d'administration

Ce type de file d'attente reçoit les messages d'administration MQSeries Everyplace. Vous pouvez également (facultatif) configurer une file d'attente d'administration de réponses pour recevoir les réponses aux messages administratifs émis par le système MQSeries Everyplace. Les files d'attente d'administration ne gèrent pas l'administration, elles traitent les messages qui encapsulent les détails administratifs.

File d'attente de pont MQSeries

Il s'agit d'une forme spéciale de file d'attente éloignée, avec définition sur une passerelle et file d'attente cible sur un gestionnaire de files d'attente MQSeries. Cette forme de file d'attente fournit un chemin entre les environnements MQSeries Everyplace et MQSeries. Des fonction Transformation permettent d'effectuer les éventuels reformatages de données ou de message requis. Une fonction Transformation élémentaire est fournie avec MQSeries Everyplace ; les programmeurs sont supposés la personnaliser pour l'adapter à leurs besoins propres.

MQSeries Everyplace stocke les données en toute sécurité dans les files d'attente, vérifiant que les données sont physiquement inscrites sur le support et non simplement mises en mémoire tampon par le système d'exploitation. Toutefois, MQSeries Everyplace ne consigne pas séparément les modifications aux messages et aux files d'attente. Si vous devez effectuer une restauration suite à une défaillance du support, vous devez faire appel à des solutions matérielles, telles que le recours à des systèmes disque RAID. Alternative possible : mapper la file d'attente sur un espace de stockage restaurable, tel que certains sous-systèmes de base de données.

MQSeries Everyplace n'impose pas qu'un gestionnaire de files d'attente ait des files d'attente définies. Mais, si besoin est, quatre files d'attente système sont prévues :

- **AdminQ** : requise pour la réception des messages administratifs
- **AdminReplyQ** : utilisée en option pour la réception des réponses aux messages administratifs
- **DeadLetter** : utilisée pour stocker les messages qui ne peuvent pas être délivrés
- **SYSTEM.DEFAULT.LOCAL.QUEUE** : file d'attente qui porte le même nom que la file d'attente système obligatoire sur les serveurs MQSeries

Les propriétés des files d'attente sont recensées dans le tableau 7. Notez qu'elles ne sont pas toutes applicables à tous les types de file d'attente.

Tableau 7. Propriétés des files d'attente

Propriété	Explication
Admin_Class	Classe de file d'attente
Admin_Name	Nom de file d'attente Ascii
Queue_Active	Indique que la file d'attente est active
Queue_AttRule	Classe de règle contrôlant les opérations de sécurité
Queue_Authenticator	Classe de l'authentificateur
Queue_BridgeName	Nom du pont MQ propriétaire
Queue_ClientConnection	Nom de connexion client
Queue_CreationDate	Date de création de la file d'attente
Queue_Compressor	Classe du module de compression
Queue_Cryptor	Classe du module de chiffrement
Queue_CurrentSize	Nombre de messages en file d'attente
Queue_Description	Description unicode
Queue_Expiry	Heure d'expiration des messages
Queue_FileDesc	Emplacement de stockage de la file d'attente
Queue_MaxMsgSize	Longueur maximale des messages autorisée dans la file d'attente
Queue_MaxQSize	Nombre maximal de messages autorisés
Queue_Mode	Synchrone ou asynchrone
Queue_MQMgr	Proxy du gestionnaire de files d'attente MQSeries
Queue_Priority	Priorité à utiliser pour les messages (sauf si autre valeur spécifiée au niveau du message)
Queue_QAliasNameList	Autres noms de la file d'attente
Queue_QMgrName	Gestionnaire de files d'attente propriétaire de la file d'attente réelle
Queue_RemoteQName	Nom de la zone MQSeries éloignée
Queue_Rule	Classe de règle pour les opérations sur files d'attente
Queue_TargetRegistry	Type de registre cible

Tableau 7. Propriétés des files d'attente (suite)

Propriété	Explication
Queue_Transporter	Classe du transport
Queue_Transformer	Classe de la fonction Transformation

Les fonctions administratives servent à créer et supprimer des files d'attente, et à consulter ou modifier leurs propriétés.

Les files d'attente ne sont pas simplement des espaces de stockage des messages. Des files d'attente transformées en sous-classes peuvent être utilisées dans des scénarios d'applications de contrôle de processus. Par exemple, l'objet file d'attente peut contrôler directement une valeur. Un message du type approprié entraîne l'ouverture d'une soupape, le changement du volume du flux, etc. L'application n'a plus à extraire les messages de la file d'attente et à exécuter l'action, c'est l'objet file d'attente lui-même qui contrôle l'action. D'autres files d'attente peuvent, par exemple, mettre à jour des feuilles de calcul ou convertir du texte en fichier vocal. Cette technique a l'avantage de préserver les garanties de sécurité des files d'attente tout comme la garantie de messagerie. MQSeries Everyplace assure alors la livraison garantie unique des messages et, associé à un authenticateur et à un module de chiffrement, garantit que seul l'expéditeur habilité du message peut envoyer ce type de messages, le contenu étant hautement sécurisé pendant le transit. Aucune application n'est autorisée à accéder à la file d'attente, aucun accès n'étant par ailleurs requis.

Gestionnaire de files d'attente

Le gestionnaire de files d'attente MQSeries Everyplace donne à l'application accès aux messages et aux files d'attente, et contrôle les éventuels canaux. Dans MQSeries Everyplace version 1.0, un seul gestionnaire de files d'attente peut être actif sur une machine virtuelle Java (JVM) à un moment donné. S'il existe plusieurs JVM sur une machine, il peut y avoir autant de gestionnaires de files d'attente que de JVM. Les gestionnaires de files d'attente sont identifiés par leur nom, qui doit être globalement unique⁵ et par une chaîne de caractères ASCII de taille non limitée. ⁶Les gestionnaires de file d'attente peuvent être configurés avec ou sans mise en file d'attente locale. Tous les gestionnaires de file d'attente prennent en charge les opérations de messagerie synchrones ; un gestionnaire de file d'attente avec mise en file d'attente locale prend également en charge la livraison des messages en mode asynchrone.

Les caractéristiques de la livraison des messages en mode asynchrone et synchrone sont très différentes et leurs conséquences également.

5. Cette restriction n'est pas appliquée par MQSeries Everyplace ou MQSeries, mais l'existence de noms de gestionnaire de files d'attente en double peut entraîner la livraison des messages à un gestionnaire de files d'attente incorrect.

6. Pour une meilleure interopérabilité il est conseillé toutefois de respecter les conventions de dénomination MQSeries, et de limiter les noms à 48 caractères. Le système de fichiers utilisé peut également avoir une incidence sur la longueur.

Livraison asynchrone des messages⁷

Avec la livraison asynchrone des messages, l'application transmet le message à MQSeries Everyplace pour qu'il le communique à une file d'attente éloignée. Un retour est renvoyé à l'application dès l'insertion effectuée. MQSeries Everyplace stocke temporairement le message en local jusqu'à ce qu'il puisse être livré. La livraison peut être échelonnée, MQSeries Everyplace en étant le garant. Ce mode opératoire assure une livraison garantie unique. Pour plus d'informations, reportez-vous à la section «Livraison asynchrone des messages» à la page 33.

Livraison synchrone des messages

Le mode synchrone peut être utilisé pour adresser :

- les files d'attente cible sur un gestionnaire de files d'attente MQSeries Everyplace routé via un réseau MQSeries Everyplace
- les files d'attente cible sur un gestionnaire de files d'attente MQSeries directement connecté à une passerelle MQSeries Everyplace
- les files d'attente cible sur un serveur MQSeries connecté indirectement.

En mode synchrone, l'application place le message dans MQSeries Everyplace pour sa livraison. MQSeries Everyplace contacte en mode synchrone la file d'attente éloignée cible et y place le message. La livraison effectuée, MQSeries Everyplace revient à l'application.

Le contact avec le gestionnaire de files d'attente éloigné peut entraîner le routage par MQSeries Everyplace via des unités et/ou passerelles intermédiaires. Pour plus d'informations, reportez-vous à la section «Livraison synchrone des messages» à la page 34.

La livraison asynchrone de messages signifie ainsi que l'application locale transmet le message à MQSeries Everyplace et que sa transmission à partir du gestionnaire de files d'attente local incombe à MQSeries Everyplace. Il apparaît donc que le réseau et/ou l'application réceptrice ne doit pas être forcément disponible. Le moment de livraison effective du message est inconnu de l'application expéditrice. La livraison synchrone d'un message requiert que le réseau soit actif, mais l'application expéditrice est informée de sa transmission à la file d'attente de l'application destinataire. Dans les deux cas (transmission asynchrone ou synchrone), la disponibilité de l'application réceptrice n'est pas requise.

Les propriétés d'un gestionnaire de files d'attente local indiquent le mode de gestion local des files d'attente. Le gestionnaire doit également intégrer une définition de connexion pour chacun des gestionnaires de files d'attente distants avec lequel il doit entrer en contact. Ces définitions sont de ce fait parfois appelées définitions du gestionnaire de files d'attente éloigné. Elles peuvent contenir toutes les informations requises pour une communication directe entre les gestionnaires de files d'attente (adresse réseau comprise) ou indiquer simplement que la communication est indirecte,

7. MQSeries Everyplace ne fait pas la distinction entre les modes persistant et non-persistant proposés par MQSeries, le mode persistant étant seul pris en charge. Toutefois, au besoin, le choix d'un espace de stockage de sauvegarde de la file d'attente permet un compromis entre performances et reprise sur incident.

transitant par un gestionnaire de files d'attente intermédiaire. Dans ce dernier cas, seul le nom du gestionnaire de files d'attente du tronçon suivant est requis.

Les propriétés des gestionnaires de files d'attente sont répertoriées dans le tableau 8 et dans le tableau 9.

Tableau 8. Propriétés d'un gestionnaire de files d'attente local

Propriété	Explication
QMgr_ChnlAttrRules	Règles d'attribut de canal
QMgr_ChnlTimeout	Délai d'expiration du canal
Admin_Class	Classe du gestionnaire de files d'attente
QMgr_Description	Description unicode
Admin_Name	Nom du gestionnaire de files d'attente
QMgr_Rules	Classe de règle pour les opérations du gestionnaire de files d'attente

Tableau 9. Propriétés de connexion (gestionnaire de files d'attente éloigné)

Propriété	Explication
Con_Adapter	Descripteur du fichier de l'adaptateur
Con_AdapterOptions	Options de l'adaptateur (utilisation d'un historique, par exemple)
Con_AdapterParm	Données ASCII à utiliser par un adaptateur (nom de servlet, par exemple)
Con_Aliases	Autres noms du gestionnaire de files d'attente
Con_Channel	Type de canal à utiliser par la connexion
Con_Description	Description unicode
Queue_QMgrName	Propriétaire de la définition
Admin_Name	Nom du gestionnaire de files d'attente

Une définition de connexion peut prendre en charge plusieurs adaptateurs.

Opérations du gestionnaire de files d'attente

Les gestionnaires de files d'attente gèrent les opérations de messagerie et, éventuellement, les files d'attente. Les applications accèdent aux messages via des opérations exécutées sur un gestionnaire de files d'attente. Sauf si un filtre est spécifié, le premier message disponible de la file d'attente est extrait. Un filtre est un objet zone qui permet de vérifier une relation d'égalité. Toute zone du message peut être sélectionnée pour une recherche sélective. L'opération de récupération ("get"), comme les opérations d'envoi et d'extraction de message, peut être affectée d'un objet attribut, utilisable pour le codage et le décodage d'un message.

Dans MQSeries Everyplace, comme dans MQSeries, l'opération "get" est normalement une opération destructive. S'il est impératif que la messagerie soit garantie entre MQSeries Everyplace et l'application, vous devez utiliser la séquence get suivi d'une

méthode de confirmation. Un get est d'abord émis avec un ID de confirmation (sa valeur étant choisie par l'application) - cette opération récupère le message mais le cache dans la file d'attente au lieu de le supprimer immédiatement. Une opération de confirmation, spécifiant l'UID message d'origine, indique ensuite que l'opération get a abouti pour l'application et le message est alors supprimé. Si l'opération get échoue, le message peut ainsi être restauré. Le comportement des opérations "put" est similaire.

En spécifiant l'UID, les messages peuvent être supprimés d'une file d'attente sans être extraits.

Si la lecture non destructive est requise, il est possible de parcourir les files d'attente à la recherche de messages (éventuellement sous contrôle d'un filtre). Cette exploration extrait tous les objets message correspondant au filtre, mais les laisse dans la file d'attente. L'exploration sous verrou est également prise en charge. Cette fonction verrouille les messages qui correspondent au filtre dans la file d'attente. Le verrouillage peut être individuel ou groupé selon un filtre. Cette opération renvoie un id de verrou. Les messages verrouillés ne peuvent être récupérés ou supprimés que si l'UID de verrou est fourni. Une option d'exploration permet de spécifier si l'intégralité des messages ou seuls les UID doivent être renvoyés.

Les applications peuvent attendre, pendant un laps de temps défini, l'arrivée des messages dans une file d'attente. Il est également possible de définir un filtre pour identifier les messages intéressants, ou encore de spécifier un id de confirmation. Autre possibilité : les applications peuvent écouter les événements message MQSeries Everyplace, là encore avec un filtre en option. Les programmes d'écoute sont avertis de l'arrivée des messages dans les files d'attente.

Le tableau 10 indique les opérations actives sur les différents types de file d'attente.

Tableau 10. Opérations de messagerie sur les file d'attente MQSeries Everyplace

	File d'attente locale	File d'attente éloignée ¹	
		Synchrone	Asynchrone
Parcourir (±verrou, ±filtre)	Oui	Oui	
Supprimer	Oui	Oui	
Get (±filtre)	Oui	Oui	
Ecouter (±filtre)	Oui		
Put	Oui	Oui	Oui
Attendre (±filtre)	Oui	Oui	
Remarques :			
1. L'opération d'attente à distance synchrone est implémentée via un pôle de la file d'attente éloignée, de sorte que le délai d'attente réel soit un multiple du délai d'interrogation.			
2. ¹ Le pont MQSeries Everyplace MQSeries fourni avec MQSeries Everyplace Version 1.0 ne prend en charge que l'opération "put".			

Les gestionnaires de files d'attente peuvent en outre charger des applications (classes) immédiatement après le démarrage ; de même, ils peuvent mettre fin aux applications au moment de l'arrêt. Les gestionnaires de files d'attente émettent des événements reflétant l'état ou l'erreur. Par défaut, ces événements sont consignés dans le journal des événements.

Administration

L'administration fournit des fonctions permettant de configurer et de gérer les ressources MQSeries Everyplace, telles que files d'attente et connexions. Les fonctions relatives aux messages sont de la responsabilité des applications. L'administration est activée via une interface qui gère la génération et la réception des messages administratifs, conçue de sorte que l'administration locale et éloignée soit gérée de la même manière. Les demandes sont envoyées à la file d'attente d'administration du gestionnaire de files d'attente cible et les réponses peuvent être réceptionnées si nécessaire. Tout programme d'application MQSeries Everyplace local ou éloigné peut créer et traiter les messages d'administration, directement ou indirectement via les méthodes du module d'assistance. Les messages d'administration peuvent également être générés indirectement via MQSeries Everyplace Explorer⁸, outil de gestion qui propose une interface utilisateur graphique pour l'administration du système.

La file d'attente d'administration ne "sait" pas administrer les ressources individuelles. Cette connaissance est encapsulée dans chaque ressource et dans le message d'administration correspondant.

Messages d'administration

Les messages d'administration étendent l'objet message MQSeries Everyplace de base. Le tableau 11 recense les classes de messages fournies pour l'administration des ressources MQSeries Everyplace. Ces messages d'administration de base peuvent être définis comme sous-classes pour fournir l'administration d'autres objets ; par exemple, une file d'attente de type différent peut être gérée via une sous-classe de MQeQueueAdminMsg. Le pont MQSeries Everyplace vers to MQSeries utilise des sous-classes de MQeAdminMsg comme suit.

Tableau 11. Classes de message d'administration

Classe de message d'administration	Utilisation
MQeAdminMsg	Classe abstraite utilisée comme base de tous les messages d'administration
MQeQueueManagerAdminMsg	Administration des gestionnaires de files d'attente
MQeQueueAdminMsg	Administration des files d'attente locales
MQeRemoteQueueAdminMsg	Administration des files d'attente éloignées
MQeAdminQueueAdminMsg	Administration de la file d'attente d'administration
MQeHomeServerQueueAdminMsg	Administration des files d'attente du serveur local

8. MQSeries Everyplace Explorer n'est pas intégré à la Version 1.0, mais peut être téléchargé à partir du site de téléchargement de logiciels MQSeries sur le Web (<http://www-4.ibm.com/software/ts/MQSeries/>).

Tableau 11. Classes de message d'administration (suite)

Classe de message d'administration	Utilisation
MQeStoreAndForwardQueueAdminMsg	Administration des files d'attente de stockage et réacheminement
MQeConnectionAdminMsg	Administration des connexions entre gestionnaires de files d'attente
MQeClientConnectionAdminMsg	Administration d'un objet de connexion de client pont, utilisé pour la connexion à MQS
MQeListenerAdminMSg	Administration d'un objet programme d'écoute de file d'attente de transmission de pont, utilisé pour collecter les messages à partir du MQS
MQeBridgeAdminMsg	Administration d'un pont vers MQS
MQeMQBridgesAdminMsg	Administration d'une liste de ponts MQ
MQeMQMGrProxyAdminMsg	Administration de la représentation d'un pont d'un gestionnaire de files d'attente MQSeries
MQeMQBridgeQueueAdminMsg	Administration d'une file d'attente d'un pont MQ

La structure d'un message administratif dépend de sa classe spécifique, c'est-à-dire de la nature de la ressource qu'il gère et des détails de l'opération à effectuer sur cette ressource. Génériquement toutefois, les messages administratifs sont structurés comme indiqué au tableau 12.

Tableau 12. Structure générique d'un message administratif

Zones de niveau 1	Zones de niveau 2 et inférieurs	Utilisation
Admin_Action		Création, suppression, interrogation, etc.
Admin_Errors		Parent de l'objet zones
	Plusieurs zones	Informations détaillées, erreur par erreur
Admin_MaxAttempts		Nombre maximal de tentatives d'exécution de l'action administrative
Admin_Parameters		Parent de l'objet zones
	Ressource	Nom de la ressource à gérer
	Plusieurs zones	Données détaillées sur le paramètre spécifique de la classe du message et de l'action
Admin_Reason		Message texte indiquant la raison de l'échec
Msg_ReplyToQ		Nom de la file d'attente à laquelle la réponse doit être envoyée

Tableau 12. Structure générique d'un message administratif (suite)

Zones de niveau 1	Zones de niveau 2 et inférieurs	Utilisation
Msg_ReplyToQMgr		Nom du gestionnaire de files d'attente auquel la réponse doit être envoyée
Admin_RC		Code retour numérique indiquant le résultat
Msg_Style		Commande ou demande/réponse
Admin_TargetQMgr		Nom du gestionnaire de files d'attente propriétaire de la ressource cible

Il existe trois styles de messages administratifs, à savoir les commandes (datagrammes) qui indiquent une action administrative ne nécessitant pas de réponse, les demandes qui requièrent une réponse et les réponses elles-mêmes. La réponse est construite à partir d'une copie du message original ; ainsi l'expéditeur peut ajouter des zones utilisables par le destinataire.

Outre le support de base des messages administratifs, des classes auxiliaires qui encapsulent la construction du message et l'interprétation de la réponse sont proposées pour les opérations d'administration les plus courantes. Ces classes peuvent éventuellement fournir des boîtes de dialogue utilisateur, utiles pour construire des outils d'administration simples.⁹

Administration sélective

L'accès à l'administration peut être contrôlée via l'authentificateur sur la file d'attente d'administration. L'authentificateur fourni considère toutes les applications locales comme représentant le même utilisateur local et, de ce fait, leur octroie ou leur refuse à toutes l'accès à l'administration. Les applications d'administration éloignées sont contrôlées par l'appel de l'authentificateur sur le canal avant tout flux de messages d'administration. Différents utilisateurs éloignés peuvent ainsi être distingués et agréés ou non séparément. Dans tous les cas, pour tout utilisateur, l'accès à l'administration est octroyé ou refusé dans sa totalité. Pour obtenir un niveau de contrôle plus précis, (par exemple, accorder à certains utilisateurs l'accès à certaines files d'attente), il faut le programmer. Un authentificateur plus sophistiqué peut garder trace des autorisations en les associant aux identités des utilisateurs, les messages d'administration pouvant alors être traités sur la base de ces autorisations (reportez-vous à la section Sécurité). Les règles associées aux files d'attente peuvent également être exploitées pour autoriser ou interdire des actions de manière similaire (reportez-vous à la section «Règles» à la page 42).

9. Ces classes ne sont pas intégrées à la Version 1.0, mais peuvent être téléchargées à partir du site de téléchargement de logiciels MQSeries sur le Web(<http://www-4.ibm.com/software/ts/MQSeries/>).

Surveillance et actions connexes

L'administration concerne souvent d'autres opérations que la création ou la modification d'objet, par exemple, lorsqu'il s'agit de surveiller l'état du système ou de gérer les erreurs, ou encore d'informer un opérateur qu'une file d'attente est presque saturée ou de réagir lorsqu'un message entrant est trop grand pour la file d'attente cible. Tous ces aspects sont gérés dans MQSeries Everyplace via l'utilisation de règles, c'est-à-dire de classes qui sont appelées chaque fois que l'état des objets change de façon significative ou que se produisent certaines erreurs. Un jeu de classes de règles par défaut est fourni avec MQSeries Everyplace, mais ces classes sont généralement remplacées par des classes personnalisées (reportez-vous à la section «Règles» à la page 42).

Canaux dynamiques

MQSeries Everyplace établit la communication entre les gestionnaires de files d'attente d'unités et/ou passerelle via des liaisons logiques appelées canaux dynamiques. Ces canaux prennent en charge les flux bidirectionnels et sont établis à la demande par le gestionnaire de files d'attente. Asynchrone ou synchrone, la messagerie utilise les mêmes canaux, et le protocole exploité est unique dans MQSeries Everyplace. A l'opposé, MQSeries utilise généralement des canaux client pour le trafic synchrone et une paire de canaux de message pour la messagerie asynchrone bidirectionnelle. Quelques-unes des caractéristiques des canaux de message de cluster MQSeries sont semblables à celles des canaux dynamiques MQSeries Everyplace, mais il existe néanmoins des différences.

Un canal dynamique est une connexion logique entre deux gestionnaires de files d'attente, établie dans le but d'envoyer ou de recevoir des données. Plusieurs canaux concurrents peuvent coexister, y compris entre les mêmes éléments. Des caractéristiques leur sont associées : authentification, chiffrement, compression, protocole de transport utilisé, etc. Ces caractéristiques sont des fonctions complémentaires (des versions différentes peuvent être utilisées sur les différents canaux) et par conséquent chaque canal a sa propre qualité d'attributs de service de :

- **Authentificateur** : NULL ou objet authentificateur susceptible d'authentifier un canal ou un utilisateur.
- **Canal** : classe fournissant les services de transport.
- **Compresseur** : NULL ou objet module de compression susceptible d'effectuer la compression et la décompression des données.
- **Module de chiffrement** : NULL ou objet module de chiffrement susceptible d'effectuer le chiffrement et le déchiffrement.
- **Destination** : cible de ce canal, par exemple SERVER.XYZ.COM

L'authentificateur n'est généralement utilisé que lors de la configuration du canal. Les modules de compression et de chiffrement sont généralement utilisés sur tous les flux.

Le type de module de chiffrement le plus simple est MQeXorCryptor, qui chiffre les données envoyées en effectuant un OU exclusif sur les données. Ce chiffrement n'offre pas une grande sécurité, mais il rend les données inaffichables. A l'opposé, MQe3DESCryptor met en oeuvre un triple DES. Le type le plus simple de compresseur

est MQeRleCompressor, qui compresse les données en remplaçant les caractères répétitifs par un nombre. D'autres authenticateurs, modules de compression et de chiffrement sont fournis : reportez-vous au tableau 13 à la page 35.

L'établissement d'un canal fait appel aux spécifications de l'adaptateur du protocole pour déterminer les liaisons et les protocoles à utiliser. A chaque noeud intermédiaire, les définitions du canal sont explorées pour répondre aux besoins d'adressage requis par la liaison suivante. Lorsqu'il n'existe plus de définition suivante, le canal est arrêté et tout message en transit est transféré vers le gestionnaire de files d'attente.

Les canaux ne sont pas directement visibles par les applications ou les administrateurs, et ils sont établis à la demande par le gestionnaire de files d'attente. Les canaux relient les gestionnaires de files d'attente entre eux et leurs caractéristiques sont négociées et renégociées par MQSeries Everyplace en fonction des informations à transférer. Les transporteurs sont les composants MQSeries Everyplace qui exploitent les canaux pour fournir les communications au niveau file d'attente. Là encore, ces canaux ne sont pas visibles par le programmeur de l'application ou l'administrateur.

Quand la messagerie garantie est impérative, MQSeries Everyplace livre les messages à l'application une fois et une seule. Pour ce faire, il vérifie qu'un message a bien été transmis d'un gestionnaire de files d'attente à un autre, avec accusé de réception, avant de supprimer la copie à la fin de la transmission. En cas d'échec de la communication (accusé de réception non reçu), un message peut être retransmis (une seule livraison ne signifie pas une seule transmission), mais il n'est pas livré en double.

Adaptateurs

Les adaptateurs permettent de mapper MQSeries Everyplace aux interfaces des unités. Les canaux exploitent les adaptateurs du protocole pour s'exécuter sous HTTP, TCP/IP natif et autres protocoles. De même, les files d'attente exploitent les adaptateurs de stockage de zones pour établir l'interface avec un sous-système de stockage tel que la mémoire ou le système de fichiers. Les adaptateurs fournissent un mécanisme permettant à MQSeries Everyplace à la fois d'étendre son support des unités et d'autoriser la gestion des versions.

Un descripteur de fichier est une chaîne utilisée pour identifier, charger et activer un adaptateur.

Gestion de connexion commutée

Le support des unités par réseau commuté est géré par le système d'exploitation de chaque unité. Lorsque MQSeries Everyplace tente, sur une unité déconnectée, d'utiliser le réseau (pour envoyer un message par exemple) et que la pile réseau n'est pas active, le système d'exploitation démarre lui-même les services RAS (remote access services). Généralement, cette opération se matérialise par l'affichage d'un panneau proposant un profil de connexion commutée. Jusqu'à l'établissement de la connexion, le contrôle appartient au système d'exploitation. En conséquence, l'utilisateur de l'unité doit vérifier que les profils de connexion commutée appropriés sont à disposition du

système d'exploitation. Il n'existe donc pas de support explicite requis pour le réseau commuté dans les implémentations d'unité MQSeries Everyplace.

Trace

La trace est activée par un programme indépendant. Des appels sont intégrés à MQSeries Everyplace pour effectuer la trace des informations, des avertissements et des erreurs, avec les variantes du système et des utilisateurs. Les applications peuvent également appeler directement la fonction de trace et ajouter des messages de trace ou les modifier. Le programme de trace exemple fourni permet d'afficher les messages sélectionnés, de les imprimer et/ou de les transférer dans le journal des événements. D'autres programmes de trace peuvent être écrits, dotés de fonctions supplémentaires ou conçus pour formater et livrer les résultats de manière différente.

La majorité des erreurs MQSeries Everyplace sont transmises à l'application pour traitement et le gestionnaire d'exceptions de l'application peut également les transférer dans le journal de trace.

Journal des événements

MQSeries Everyplace fournit des mécanismes de consignation des événements et des interfaces, utilisables pour consigner les états, le démarrage du gestionnaire de files d'attente, par exemple. La journalisation peut être lancée et écrite par défaut dans un fichier, mais il est possible de réacheminer la sortie. Le journal des événements MQSeries Everyplace ne consigne pas les données des messages et ne peut pas être utilisé pour restaurer des messages ou des files d'attente.

Réseau MQSeries Everyplace

Les réseaux MQSeries Everyplace sont constitués d'unités et de passerelles connectées. Ils peuvent traverser plusieurs réseaux physiques et acheminer des messages de l'un à l'autre. Ils fournissent généralement un accès synchrone et asynchrone aux files d'attente avec un modèle de programmation indépendant de l'emplacement de la file d'attente.

Configurations et évolutivité

La figure 5 à la page 32 présente une sélection de topologies de réseaux MQSeries Everyplace de base. Dans cette optique, tout réseau est supposé être configuré avec des fonctions de communication synchrone et asynchrone.

La configuration la plus simple est constituée d'une unité autonome qui prend en charge les communications synchrones entre applications via des files d'attente, comme illustré en (a) ci-dessus. Plus intéressante est la configuration (b), illustrant un réseau d'égal à égal. Cette configuration suppose que les deux unités utilisent le même protocole de communication et qu'au moins l'une des deux est configurée avec des fonctions d'écoute, pour répondre aux tentatives de contact de l'autre. A l'évidence, dans cette configuration, la communication n'est possible que lorsque les deux unités se trouvent sur le même réseau. La messagerie asynchrone permet d'exploiter des

applications même si les unités ne sont pas connectées, la messagerie synchrone n'est possible que lorsque la connexion est établie.

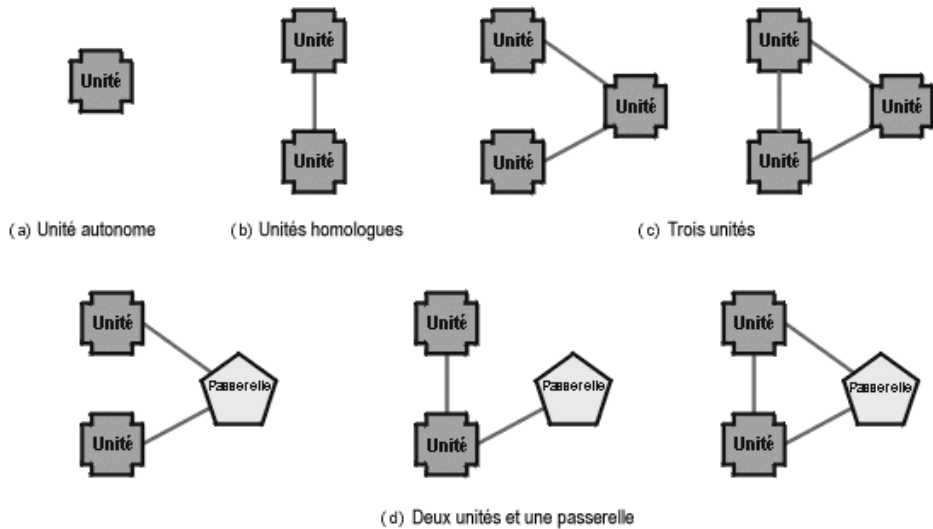


Figure 5. Réseaux MQSeries Everyplace simples

Avec trois unités, les possibilités sont nettement plus vastes. Une unité peut jouer le rôle d'unité de liaison, mais les trois unités peuvent également être interconnectées. Si elles sont configurées pour échanger des messages sur une base d'égal à égal, les considérations précédentes s'appliquent. Si la communication passe par une unité tierce, il y a davantage de souplesse. Les deux unités de communication n'ont pas besoin de passer par le même protocole de communication, il suffit que chacune opte pour un langage compris par l'unité qui sert de pont. De plus, la connectivité asynchrone permet la transmission des messages même si l'expéditeur et le récepteur ne sont jamais connectés au réseau simultanément (sous réserve que chacun soit connecté au réseau à un moment où l'unité de liaison est disponible). La communication synchrone requiert que les trois unités soient simultanément disponibles. Pour les unités qui sont fréquemment déconnectées, une configuration qui passe par un quelconque troisième noeud est la plus adaptée, sous réserve que le noeud intermédiaire soit généralement disponible.

Dans la pratique, les unités sont généralement reliées via une passerelle, comme le montrent trois des exemples de la figure 5(d). La préférence pour une passerelle comme noeud de liaison tient au fait que les passerelles acceptent plusieurs demandes de connexion entrantes simultanées. La première configuration illustrée est la plus courante. Les deux autres, quoique réalisables, sont plus inhabituelles. Dans la troisième configuration, notez qu'une seule route peut être configurée pour un gestionnaire de files d'attente éloigné spécifique et que donc, bien que deux routes semblent exister, vous devez en sélectionner une.

Pour les réseaux plus étendus, plusieurs passerelles peuvent être utilisées, chacune prenant en charge plusieurs unités. Les passerelles peuvent être interconnectées de n'importe quelle façon, mais si l'interconnexion intégrale est définie, aucune route entre les unités ne peut concerner plus de deux passerelles. La figure 6 illustre un réseau étendu.

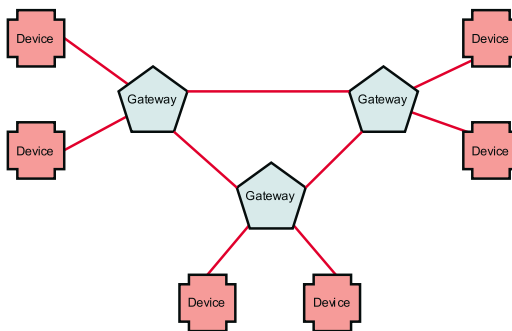


Figure 6. Réseau MQSeries Everyplace en étoile

Livraison asynchrone des messages

Lorsqu'un message est transmis en mode asynchrone à une file d'attente éloignée, l'objet message est placé logiquement dans l'espace de stockage de sauvegarde associé à la définition de cette file d'attente, avec le nom de son gestionnaire de files d'attente de destination et celui de la file d'attente, ainsi que les caractéristiques des modules de compression, de chiffrement et de l'authentificateur correspondant à la cible du message. La méthode de vidage de l'objet est appelée au moment où il est sauvegardé en mémoire persistante dans un format sécurisé, tel que défini par la file d'attente de destination. Le gestionnaire de files d'attente contrôle la livraison du message. Il identifie (ou établit) un canal doté des caractéristiques adéquates vers le gestionnaire de files d'attente pour le tronçon suivant, puis il crée (ou réutilise) un transporteur vers la file d'attente cible. Le transporteur effectue un vidage de l'objet et transmet la chaîne de caractères binaires résultante. Notez que le gestionnaire de files d'attente cible et le nom de la file d'attente ne sont pas intégrés au flux du message.

Le cas échéant, le message est chiffré et compressé sur le canal. S'il atteint le gestionnaire de files d'attente de destination, il est déchiffré et décompressé. Un nouvel objet message est créé, via la méthode de restauration de cette classe d'objet, l'objet résultant étant placé dans la file d'attente de destination. Si le message n'est pas parvenu au gestionnaire de files d'attente de destination, il est déchiffré et décompressé, puis placé dans une file d'attente de stockage et transmission avec les caractéristiques appropriées pour la transmission suivante. Dans les deux cas, il est placé en mémoire persistante dans un format sécurisé, tel que défini par la file d'attente de destination.

La livraison asynchrone des messages présente la particularité de transmettre les messages au gestionnaire de files d'attente à des tronçons intermédiaires, mis en file

d'attente en vue de la transmission suivante. Les messages sont extraits des files d'attente intermédiaires selon leur priorité, puis selon leur horodatage.

Livraison synchrone des messages

La livraison synchrone de messages est semblable à la livraison asynchrone décrite plus haut, mais l'intervention du gestionnaire de files d'attente dans les tronçons intermédiaires a lieu à un niveau bien inférieur, impliquant le transporteur et les canaux. Un canal est établi de bout en bout, à l'aide des adaptateurs définis dans les spécifications du protocole à chaque noeud intermédiaire, pour identifier la liaison suivante. A la fin de la dernière liaison, lorsqu'il n'existe plus de descripteur de fichier concerné, le message est transmis aux couches supérieures du gestionnaire de files d'attente, pour y être traité. Ainsi le noeud expéditeur ne met pas le message en file d'attente, mais le transmet par le canal, via des tronçons intermédiaires, puis le remet au gestionnaire de files d'attente destinataire pour qu'il le place dans la file d'attente cible.

La liaison dans MQSeries utilise une file d'attente de pont au niveau de la passerelle, qui convertit le message en un format MQSeries. Ce mécanisme signifie que la messagerie synchrone de type MQSeries Everyplace est possible à partir d'une unité vers MQSeries, le canal dynamique se terminant à la passerelle. Le message est livré en temps réel à partir de la passerelle, via un canal client, à un serveur MQSeries. A ce stade, sa destination peut imposer qu'il soit acheminé en mode asynchrone le long des canaux de message MQSeries.

De même, une unité qui ne peut travailler en mode synchrone peut envoyer des messages vers une file d'attente MQSeries Everyplace asynchrone, sous réserve qu'un intermédiaire approprié soit disponible.

Sécurité

MQSeries Everyplace fournit un ensemble intégré de fonctions de sécurité permettant de protéger les données des messages, lors de leur stockage local et de leur transfert.

Les fonctions de sécurité de MQSeries Everyplace assurent une protection à un triple niveau :

- Sécurité locale - protection locale des données des messages (et autres données)
- Sécurité au niveau des files d'attente - protection de messages entre le gestionnaire de files d'attente expéditeur et la file d'attente cible
- Sécurité au niveau message - protection des messages entre l'expéditeur et le destinataire

Les fonctions de sécurité MQSeries Everyplace locales et au niveau message sont utilisées en interne par MQSeries Everyplace, mais sont également à disposition des applications MQSeries Everyplace. La fonction de sécurité MQSeries Everyplace au niveau des files d'attente est un service interne.

Les trois types de fonctions de sécurité MQSeries Everyplace assurent la protection des messages via un attribut (MQeAttribute ou descendant). Selon le type, l'attribut est appliqué explicitement ou implicitement.

Chaque attribut peut contenir les objets suivants :

- Authentificateur
- Module de chiffrement
- Module de compression
- Clé
- Nom de l'entité cible

Ces objets sont utilisés différemment, selon le type de fonction de sécurité MQSeriesEveryplace, mais, dans tous les cas, la protection est appliquée lors de l'appel de l'attribut associé à un objet message. Ce qui se produit lors de l'appel d'une méthode de vidage ("dump") MQSeries Everyplace du message (par exemple, lorsque la méthode "encodeData" de l'attribut est utilisée pour chiffrer et compresser les données du message). L'abandon de la protection par les fonctions de sécurité MQSeries Everyplace a lieu lorsque la méthode "restore" du message est appelée (par exemple, lorsque la méthode 'decodeData' de l'attribut est utilisée pour décompresser et déchiffrer les données du message).

Les algorithmes pris en charge par MQSeries Everyplace version 1.0 pour l'authentification, le chiffrement et la compression sont explicités dans le tableau 13.

Tableau 13. Support de l'authentification, du chiffrement et de la compression

Fonction	Algorithme
Authentification	Basé sur Mini Certificate (dérivé du WAP forum WTLS Mini Certificate)
	Validation d'identité Windows NT/2000
Compression	LZW
	RLE
Chiffrement	Triple DES
	DES
	MARS
	RC4
	RC6
	XOR

Sécurité locale MQSeries Everyplace

Les fonctions de sécurité locales protègent localement les données des messages MQSeries Everyplace (ou les descendants MQeFields ou MQeFields). Pour ce faire, il convient de créer un attribut, avec un module de chiffrement et de compression symétrique approprié, de créer et de configurer une "clé" adéquate (par mot de passe simple ou composé) et d'associer explicitement la clé à l'attribut, puis l'attribut au

message MQSeries Everyplace. MQSeries Everyplace fournit une classe MQeLocalSecure, qui aide à la configuration des fonctions de sécurité locales, mais il incombe dans tous les cas à l'utilisateur de la fonction de sécurité locale (MQSeries Everyplace en interne ou application MQSeries Everyplace) de configurer l'attribut approprié et de gérer la clé par mot de passe simple ou composé.

Sécurité MQSeries Everyplace au niveau des files d'attente

Les fonctions de sécurité au niveau des files d'attente sont applicables aux messages synchrones et asynchrones.

Sécurité au niveau des files d'attente synchrones

L'utilisation de fonctions de sécurité fondées sur les files d'attente permet à une application de déléguer à MQSeries Everyplace tout souci concernant la sécurité des messages. Les fonctions d'authentification, de chiffrement et de compression associées aux files d'attente sont utilisées pour déterminer le niveau de sécurité requis pour protéger les flux de message (ainsi que les messages stockés en mémoire persistante).

Lorsqu'un message est sur le point d'être envoyé, les fonctions de sécurité de la file d'attente cible sont extraites du registre local. En leur absence, le gestionnaire de files d'attente tente de les retrouver à partir du gestionnaire cible et les place en mémoire cache pour les utiliser ultérieurement. S'il existe un canal vers ce gestionnaire de files d'attente, il est utilisé. A défaut, il est créé. Les attributs de la file d'attente cible sont extraits.

Selon la qualité du service requis, les attributs du canal vers le gestionnaire de files d'attente cible sont modifiés dynamiquement. Cette opération est soumise à toutes les règles établies, s'il y en a. En général, une règle autorise la progression du niveau de sécurité (par exemple, de l'absence de protection à une protection de faible niveau, ou de faible à fort). S'il est impossible de mettre à jour le canal ou que le niveau de sécurité est jugé excessif (par exemple, aucune protection requise et un canal implémentant un niveau de protection maximal), un nouveau canal est créé. Il existe un pool de canaux, réutilisés chaque fois que possible, dont les caractéristiques sont modifiées dynamiquement selon les impératifs du trafic. Les canaux devenant inutiles sont détruits automatiquement. Les messages sont toujours placés dans les files d'attente au niveau de sécurité défini par les caractéristiques de la file d'attente cible.

L'authentification a lieu au niveau du canal, pour réduire au minimum la charge par message. De même, la sécurité au niveau des files d'attente synchrones est généralement associée à des modules de chiffrement symétriques, ce qui accélère les processus de chiffrement/déchiffrement. Toutefois, dans ces cas de symétrie, MQSeries Everyplace utilise au départ le chiffrement asymétrique RSA, afin de protéger le flux requis pour établir une clé partagée pour l'expéditeur et le destinataire. Le chiffrement symétrique est utilisé ensuite pour protéger la confidentialité des données transmises. MQSeries Everyplace rend difficile toute attaque visant à déchiffrer ces données, en modifiant dynamiquement la clé sur chaque flux du canal. MQSeries Everyplace ajoutant la synthèse aux données avant leur envoi et en la régénérant et en la validant à réception.

Sécurité au niveau des files d'attente asynchrones

La transmission asynchrone des messages diffère de leur transmission synchrone, décrite plus haut, en ceci qu'il n'existe pas de garantie que la file d'attente cible soit accessible au moment de l'exécution de `putMessage`. Dans ce cas, le gestionnaire de files d'attente ne peut envoyer le message immédiatement et le place dans sa file d'attente de transmission. Le message est toutefois chiffré en fonction des caractéristiques de la file d'attente cible. Lorsqu'il peut être transmis, il est déchiffré, puis envoyé via un canal doté des caractéristiques appropriées. Les messages sont ainsi toujours protégés, même lorsqu'ils sont en attente de transmission. La transmission asynchrone des messages requiert la définition d'une file d'attente éloignée, faute de quoi il est impossible de déterminer les caractéristiques de la file d'attente cible.

En mode asynchrone, l'authentification entre l'expéditeur et le destinataire est impossible. Si l'authentification est importante, que, par exemple, le destinataire doit déterminer qui envoie le message pour l'accepter ou définir une irréfutabilité, ou que l'expéditeur souhaite s'assurer que le message ne peut être traité que par le destinataire indiqué, il convient d'utiliser les fonctions de sécurité au niveau message.

Les fonctions de sécurité au niveau des files d'attente peuvent être utilisées simultanément avec les fonctions de sécurité au niveau message, mais cela est superflu, les données des messages étant déjà protégées.

Sécurité au niveau message

Les fonctions de sécurité au niveau message assurent la protection des données des messages entre une application MQSeries Everyplace expéditrice et une application réceptrice.

La sécurité au niveau message est un service de la couche application qui requiert que l'application MQSeries Everyplace émettrice définisse un attribut de niveau message et le fournisse lors de l'exécution de `putMessage` pour placer le message dans une file d'attente cible. L'application réceptrice doit définir et transmettre un attribut de niveau message correspondant au gestionnaire de files d'attente récepteur de sorte que l'attribut soit disponible lorsque l'application appelle `getMessage` pour récupérer le message de la file d'attente cible.

A l'instar de la sécurité locale, la sécurité au niveau message exploite l'application d'un attribut sur un objet message. Le gestionnaire de files d'attente de l'application émettrice gère le `putMessage` avec la méthode "dump", laquelle se sert de la méthode "encodeData" de l'attribut pour protéger les données du message. Le gestionnaire de files d'attente de l'application réceptrice gère `getMessage` de l'application par la méthode "restore", laquelle se sert de la méthode "decodeData" de l'attribut pour restaurer les données du message d'origine.

MQSeries Everyplace propose deux autres attributs pour la sécurité au niveau message :

MQeMAttribute

Cet attribut est utilisé pour les communications inter-entreprises, lorsque la sécurisation mutuelle est strictement gérée dans la couche application et ne

requiert pas de tiers sécurisé. Tous les modules de chiffrement et de compression MQSeries Everyplace symétriques disponibles peuvent être utilisés. Comme pour la sécurité locale, la clé de l'attribut doit être prédéfinie, puis fournie avec `putMessage` ou `getMessage`. MQeAttribute fournit une méthode à la fois simple et puissante pour la protection au niveau message, permettant un chiffrement efficace pour protéger la confidentialité des messages, sans la surcharge induite par toute infrastructure PKI (public key infrastructure).

MQeMTrustAttribute

Cet attribut propose une solution plus élaborée, exploitant les signatures numériques et l'infrastructure de clé publique (PKI) par défaut. Il utilise la signature numérique/validation ISO9796 pour permettre à l'application réceptrice d'établir la preuve que le message provient effectivement de l'expéditeur supposé. Le module de chiffrement d'attribut fourni sert à protéger la confidentialité des messages. La synthèse SHA1 garantit l'intégrité des messages et le chiffrement/déchiffrement RSA assure que le message ne peut être restauré que par le destinataire indiqué. Comme avec MQeMAttribute, tous les modules de chiffrement et de compression MQSeries Everyplace symétriques disponibles peuvent être utilisés. Sélectionnés pour l'optimisation de la taille, les certificats utilisés sont des mini-certificats basés sur le certificat WTLS proposé par la spécification WAP forum WTLS. La disponibilité mutuelle des informations requises pour authentifier (valider les signatures) et pour chiffrer/déchiffrer est fournie via l'infrastructure MQSeries Everyplace par défaut.

Un message protégé par MQeMTrustAttribute a généralement le format :

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

où :

RSA-enc	chiffré par RSA avec la clé publique du destinataire indiqué
SymKey	clé symétrique pseudo-aléatoire générée
SymKey-enc	chiffrée symétriquement avec SymKey
Data	données du message
DataDigest	synthèse des données du message
DigSignature	signature numérique de l'expéditeur des données du message

La sécurité au niveau message est indépendante de la sécurité au niveau des files d'attente.

Registre

Le registre est le principal espace de stockage des informations concernant le gestionnaire de files d'attente. Il en existe un par gestionnaire de files d'attente. Chaque gestionnaire utilise le registre pour y conserver :

- Ses données de configuration
- Les définitions des files d'attente

- Les définitions des files d'attente éloignées
- Les définitions du gestionnaire de files d'attente éloigné
- Les données utilisateur (dont les informations de sécurité dépendant de la configuration)

L'accès au registre, protégé par PIN, est normalement réservé à l'utilisateur légitime du gestionnaire de files d'attente, mais il existe une option configurable qui permet de passer outre, pour des utilisateurs plus soucieux de la taille des empreintes que de la sécurité.

Entités MQSeries Everyplace authentifiables

La sécurité au niveau des files d'attente, qui utilise l'authentification mutuelle sur la base de mini-certificats et la protection au niveau message, qui utilise les signatures numériques, ont amené à définir le concept d'"entité authentifiable". Dans le cas d'authentification mutuelle, il paraît normal de penser authentification entre deux utilisateurs (personnes), mais en général, la fonction de messagerie n'intègre pas le concept d'utilisateur. Ce concept est généralement géré au niveau application, c'est-à-dire par l'utilisateur des services de messagerie. MQSeries Everyplace fait délibérément abstraction du concept de "cible de l'authentification" (représentant un utilisateur) au profit du concept d'"entité authentifiable". Ce qui n'exclut pas la possibilité qu'une entités authentifiable soit une personne, mais il s'agit alors d'un mappage sélectionné par une application. En interne, MQSeries Everyplace définit comme entité authentifiable tous les gestionnaires de files d'attente qui peuvent être soit l'origine soit la cible de services dépendant de mini-certificats. En outre, MQSeries Everyplace définit également comme entité authentifiable les files d'attente configurées pour utiliser les authenticateurs basés sur les mini-certificats. Ainsi, les gestionnaires de files d'attente qui prennent en charge ces services peuvent être associés à une entité authentifiable, le gestionnaire de files d'attente, ou à un ensemble d'entités authentifiables, le gestionnaire de files d'attente et toutes les files d'attente qui utilisent un authenticateur basé sur les certificats.

Registre privé et accréditations

Pour être utile, chaque entité authentifiable doit être dotée de ses propres accréditations. Ce qui pose deux questions. La première consiste à savoir comment effectuer l'enregistrement pour obtenir les accréditations et la seconde, de déterminer où gérer les accréditations de façon sécurisée. Normalement, ces questions sont plus difficiles à résoudre que ne le sont les techniques de chiffrement sous-jacentes. MQSeries Everyplace fournit des services par défaut qui permettent aux entités authentifiables de s'enregistrer automatiquement, un registre privé (descendant du registre de base) pour la gestion sécurisée des accréditations privées d'une entité authentifiable et un registre public (également descendant du registre de base) pour la gestion de l'ensemble des accréditations publiques. Le registre privé fournit le registre de base avec plusieurs des qualités d'un jeton sécurisé ou chiffré. Ce peut être, par exemple, un référentiel sécurisé pour des objets publics tels que les mini-certificats, et des objets privés tels que les clés privées. Il fournit un mécanisme qui réserve aux seuls utilisateurs autorisés l'accès aux objet privés. Il assure également le support de services (par exemple, signature numérique, déchiffrement RSA) de telle sorte que les objets privés ne quittent jamais le registre privé. En fournissant une interface commune,

il cache le support d'unité sous-jacent, actuellement restreint au système de fichiers local, mais pourrait être étendu pour correspondre aux jetons des portables dans un proche avenir.

Enregistrement automatique

MQSeries Everyplace fournit des services par défaut qui prennent en charge l'enregistrement automatique. Ces services sont automatiquement déclenchés lorsqu'une entité authentifiable est configurée, par exemple au démarrage d'un gestionnaire de files d'attente ou à la définition d'une nouvelle file d'attente. Dans les deux cas, l'enregistrement est déclenché et des accréditations sont créées et stockées dans le registre privé de l'entité privée. Les étapes d'enregistrement automatique comprennent la génération d'une nouvelle paire de clés RSA, la protection et la sauvegarde de la clé privée dans le registre privé et l'intégration de la clé publique dans une demande de "new certificate" adressée au serveur de mini-certificats par défaut. Si le serveur de certificats est configuré et disponible, il renvoie les nouveaux mini-certificats de l'entité authentifiable, avec son propre mini-certificat, et ceux-ci, avec la clé privée protégée, sont stockés dans le registre privé de l'entité authentifiable comme ses nouvelles accréditations. Alors que l'enregistrement automatique fournit un mécanisme simple pour établir les accréditations d'un entité authentifiable, la protection au niveau message (MQeMTrustAttribute, voir plus haut) impose l'accès à la clé publique (mini-certificat) du destinataire désigné.

Registre public et réplication de certificat

MQSeries Everyplace fournit des services par défaut qui permettent le partage des accréditations publiques d'une entité authentifiable (mini-certificats) entre les composants MQSeries Everyplace. Ce sont des conditions préalables pour la sécurité au niveau message MQeMTrust. Le registre public MQSeries Everyplace fournit un référentiel publiquement accessible pour les mini-certificats. Cela est analogue au répertoire téléphonique personnel sur les téléphones mobiles, la différence étant que les numéros de téléphone sont ici un ensemble de mini-certificats des entités authentifiables le plus souvent contactées. Le registre public n'est pas complètement passif dans ses services. S'il fait l'objet d'un accès pour fournir un mini-certificat qu'il ne détient pas, et sous réserve qu'il soit configuré avec un composant serveur local valide, le registre public tente automatiquement de récupérer le mini-certificat demandé dans le registre public du serveur local. Ces services peuvent être utilisés pour fournir un service de réplication de mini-certificats automatisé intelligent, qui assure la disponibilité du mini-certificat requis au moment requis.

Utilisation des services de registre par les applications

Dans la mesure où le gestionnaire de files d'attente MQSeries Everyplace est conçu pour exploiter les avantages des services de registres privés et publics, l'accès à ces services n'est pas limité. Les solutions MQSeries Everyplace peuvent souhaiter définir et gérer leur propres entités authentifiables, les utilisateurs par exemple. Les services de registre privé peuvent dans ce cas être utilisés pour enregistrer automatiquement et gérer les accréditations des nouvelles entités authentifiables, les services de registre public assurant alors la mise à disposition des accréditations publiques lorsque cela est nécessaire. Toutes les entités authentifiables enregistrées peuvent être utilisées comme expéditeur ou récepteur des services au niveau message, protégés via MQeMTrustAttribute

Service d'émission de mini-certificats par défaut

MQSeries Everyplace fournit un service d'émission de mini-certificats par défaut, qui peut être configuré pour répondre aux demandes d'enregistrement automatique de registres privés. Grâce aux outils fournis par MQSeries Everyplace, une solution peut configurer et gérer un service d'émission de mini-certificats, qui délivre des mini-certificats à un ensemble strictement contrôlé de noms d'entités. Les caractéristiques de ce service d'émission sont les suivantes :

- Gestion de l'ensemble des entités authentifiables enregistrées
- Livraison de mini-certificats (basés sur les mini-certificats WAP WTLS)
- Gestion du référentiel de mini-certificats

Les outils fournis par MQSeries Everyplace permettent à un administrateur du service d'émission de mini-certificats d'autoriser la livraison de mini-certificat à une entité spécifique, en enregistrant son nom et son adresse et en définissant un PIN de demande de certificat à usage unique. Cette opération est normalement effectuée un fois que le contrôle hors ligne a vérifié l'authenticité du demandeur. Le PIN de demande de certificat est posté à l'utilisateur désigné (de la même manière, par exemple, que les PIN de carte bancaire sont postés aux utilisateurs à l'émission d'une nouvelle carte). L'utilisateur du registre privé (par exemple, l'application MQSeries Everyplace ou le gestionnaire de files d'attente MQSeries Everyplace) peut alors être configuré pour fournir ce PIN de demande de certificat au moment du démarrage. Lorsque le registre privé déclenche l'enregistrement automatique, le service d'émission de mini-certificats valide la nouvelle demande de certificat résultante (sur la base d'une correspondance entre le nom d'entité présentée et le PIN de demande de certificat avec les valeurs préenregistrées), délivre le nouveau mini-certificat et réinitialise le PIN de demande du certificat enregistré de sorte qu'il ne puisse pas être réutilisé. Toutes les demandes d'enregistrement automatique de nouveaux mini-certificats sont traitées dans un canal sécurisé.

L'ensemble des mini-certificats délivrés par un service d'émission de mini-certificats est conservé dans le registre propre au service. Lorsqu'un mini-certificat est ré-émis (par exemple, suite à une expiration), le mini-certificat arrivé à expiration est archivé.

Interface de sécurité

Une interface en option peut être implémentée par un gestionnaire de sécurité personnalisé. Les méthodes permettent à ce gestionnaire d'accepter ou de rejeter les demandes associées à :

- L'ajout ou le retrait d'alias de classe
- La définition d'adaptateurs
- Le mappage de descripteurs de fichier
- Le traitement des commandes de canaux

Configuration et personnalisation

Règles

Les règles sont des classes Java utilisées pour personnaliser le comportement de MQSeries Everyplace lorsque se produisent différents changements d'état. Les règles par défaut sont fournies lorsque cela est nécessaire, mais peuvent être remplacées par des règles spécifiques de l'application - ou de l'installation - pour mieux répondre aux besoins du client. Les types de règles pris en charge varient selon leur mode de déclenchement, et non selon leur action. Les règles comportent une logique et sont donc susceptibles d'effectuer de nombreuses fonctions.

Règles d'attribut

Cette classe de règles prend le contrôle à chaque tentative de changement d'état, par exemple, un changement de l'un des éléments suivants :

- Authentificateur
- Module de compression
- Module de chiffrement

La règle autorise ou refuse le changement.

Règles du pont MQSeries

Ces classes de règles prennent le contrôle lorsque le code MQSeries Everyplace vers le pont MQSeries subit un changement d'état. Il existe une classe de règles de pont distincte pour déterminer ce qui suit :

- Que faire d'un message issu de MQSeries lorsqu'un programme d'écoute ne peut le livrer sur MQSeries Everyplace. Par exemple, lorsque le message est trop volumineux et qu'il n'existe pas de file d'attente.
- Les objets d'état administrés par le pont qui doivent démarrer dès que le serveur est instancié.
- Que faire lorsque le pont découvre une anomalie au niveau de Sync Q sur MQSeries (espace de stockage persistant utilisé pour la reprise en cas de panne). La règle par défaut se contente de signaler le problème.
- Comment convertir un message MQSeries Everyplace en message MQ, et vice versa. Les fonctions Transformation susceptibles d'effectuer la conversion entre messages MQSeries Everyplace et messages MQ ne sont pas dérivées de classes MQeRule. Elles doivent implémenter l'interface MQeTransformerInterface. A cette différence près, les fonction Transformation ont le même rôle que les règles et sont appelées lorsqu'un message doit être converti.

Règles du composeur RAS

Cette classe de règles prend le contrôle à chaque changement d'état du composeur RAS, par exemple :

- Si la connexion au numéro appelé ne s'établit pas
- Si le seuil d'erreurs est dépassé

- Si une numérotation est tentée et que seuls certains types de connexions sont utilisables, selon l'heure de la journée. Par exemple, si l'usage du téléphone est réservé aux heures creuses.

Règles de file d'attente

Cette classe de règles prend le contrôle à chaque changement d'état de la file d'attente associée, par exemple :

- Ajout d'un message à la file d'attente. Par exemple pour vérifier s'il y a dépassement d'un seuil (nombre de messages, taille de message, priorité incorrecte)
- Affectation ou modification des caractéristiques de la file d'attente
- Ouverture ou fermeture de la file d'attente
- File d'attente sur le point d'être supprimée.

Règles du gestionnaire de files d'attente

Cette classe de règles prend le contrôle à chaque changement d'état du gestionnaire de files d'attente, par exemple :

- Ouverture du gestionnaire de files d'attente. Par exemple, au démarrage d'une horloge en arrière-plan pour permettre l'exécution d'actions programmées.
- Fermeture du gestionnaire de files d'attente. Par exemple, pour arrêter l'horloge en arrière-plan.
- Ajout d'une nouvelle file d'attente.

Styles de connexion

MQSeries Everyplace peut prendre en charge les opérations client-serveur et/ou d'égal à égal.¹⁰ Un client est capable d'initier une communication avec un serveur ; un serveur ne peut que répondre aux demandes initiées par un client. Dans une opération d'égal à égal, les deux homologues peuvent initier les flux dans l'un ou l'autre sens. Ces styles de connexion supposent la disponibilité et l'activation de composants MQSeries Everyplace différents. Les composants impliqués sont les suivants :

- **Programme d'écoute de canal** : est à l'écoute des demandes de connexion entrantes.
- **Gestionnaire de canal** : prend en charge les canaux de communication concurrents logiques multiples entre points terminaux.
- **Gestionnaire de files d'attente** : prend en charge les applications via des fonctions de messagerie et de mise en file d'attente.

Le tableau 14 à la page 44 illustre les relations entre ces composants et le style de connexion. Le style de connexion client/serveur décrit la situation dans laquelle MQSeries Everyplace peut opérer indifféremment en mode client ou serveur. L'option servlet décrit le cas où MQSeries Everyplace est configuré comme servlet HTTP, le serveur HTTP lui-même étant responsable de l'écoute des demandes de connexion entrantes.

10. Dans cette section, les termes "client" et "serveur" font référence à leur acception courante et non à la sémantique propre à MQSeries.

Tableau 14. Styles de connexion

	Gestionnaire de files d'attente	Gestionnaire de canal	Programme d'écoute de canal
Client	Oui		
Client/serveur	Oui	Oui	Oui
Homologue	Oui		
Serveur	Oui	Oui	Oui
Servlet	Oui	Oui	

L'utilisation d'un gestionnaire de canal MQSeries Everyplace ou d'un programme d'écoute de canal MQSeries Everyplace détermine, au niveau de la licence, si une instance de MQSeries Everyplace est une passerelle.

Les applications MQSeries Everyplace ne sont pas directement avisées du style de connexion utilisé par les gestionnaires de files d'attente. Le style est toutefois significatif en ceci qu'il a une incidence sur les ressources disponibles pour les deux parties, sur les gestionnaires de files d'attente susceptibles de se connecter à d'autres, sur l'empreinte MQSeries Everyplace au niveau d'une unité ou d'une passerelle, et sur les connexions admises à coexister.

Connexion d'égal à égal

Un canal d'égal à égal comprend les fonctions d'un gestionnaire de canal et un programme d'écoute pour un seul canal. A la création d'un canal d'égal à égal entre deux gestionnaires de files d'attente, l'un de ceux-ci doit jouer le rôle de programme d'écoute et l'autre celui d'initiateur de la connexion. Un gestionnaire de files d'attente connecté en mode d'égal à égal peut initier plusieurs connexions d'égal à égal avec d'autres gestionnaires de files d'attente, mais ne peut répondre qu'à une demande de connexion entrante. Il doit ensuite attendre la fermeture du canal d'égal à égal avant de répondre à une autre demande. Via un canal d'égal à égal quelconque, les deux gestionnaires de files d'attente concernés peuvent tous deux initier des actions. Ainsi, par exemple, les applications de chacun des gestionnaires ont accès aux files d'attente de l'autre.

Les canaux d'égal à égal peuvent être inutilisables à travers des pare-feu, dans la mesure où la cible de la demande de connexion entrante peut être rejetée par le pare-feu.

Connexion client-serveur

Les canaux standard, utilisés pour le style de connexion client-serveur, sont dépourvus de fonctions d'écoute, et dépendent d'un programme d'écoute indépendant, au niveau du serveur, lequel serveur requiert un gestionnaire de canal pour gérer les canaux concurrents multiples. Le client initie la demande de connexion et le serveur répond. Un serveur est généralement capable de gérer plusieurs demandes entrantes issues des clients. Via un canal standard, le client a accès à des ressources sur le serveur. Si une application sur le serveur requiert un accès synchrone aux ressources sur le client, il est nécessaire de configurer un second canal, où les rôles sont inversés. Toutefois, les

canaux standard étant eux-mêmes bidirectionnels, les messages destinés à un client issus de la file d'attente de transmission de son serveur sont livrés via le canal (client-serveur) standard qui est lancé.

Un client peut être simultanément client de plusieurs serveurs. Notez qu'il n'est pas besoin d'un gestionnaire de canal pour prendre en charge cette configuration, car les gestionnaires de canaux gèrent plusieurs canaux internes.

Le style de connexion client-serveur est généralement adapté aux transmissions via un pare-feu dans la mesure où la cible de la connexion entrante est normalement identifiée comme acceptable par le pare-feu.

Styles de connexion multiple

Un gestionnaire de files d'attente peut à lui seul être capable d'initier des connexions d'égal à égal ou client/serveur, et de répondre à l'un ou l'autre en tant que serveur ou en tant qu'homologue. Dans ce cas, le programme d'écoute du canal homologue et celui du canal standard doivent être dotés de numéros de port distincts.

Classes

MQSeries Everyplace propose un choix de classes pour certaines fonctions pour que le comportement de MQSeries Everyplace puisse être personnalisé et réponde aux conditions spécifiques des applications. Parfois, les interfaces avec les classes sont documentées, de sorte qu'il est possible de développer d'autres alternatives. Le tableau 15 recense les possibilités. Les classes peuvent être identifiées explicitement ou via des alias.

Tableau 15. Options de classes

Classe	Alternative proposée	Interfaces documentées
administration	non	oui
Authentificateurs	oui	non
Adaptateur de communications	oui	oui
Style de communication	oui	non
Modules de compression	oui	non
Modules de chiffrement	oui	non
Journal des événements	exemple fourni	oui
Messages	non	oui
Stockage de file d'attente	oui	non
Règles	classes par défaut fournies	oui
Trace	exemples fournis	oui

Chargement des applications

Lorsque MQSeries Everyplace est configuré pour opérer en tant que client (ou homologue), le chargement de toute application dans le JVM incombe à l'application initiatrice. A cet effet, vous pouvez utiliser les fonctions Java standard ou le chargeur de

classe intégré à MQSeries Everyplace. Ainsi, plusieurs applications peuvent être exécutées sur un seul gestionnaire de files d'attente dans le même JVM. Il est également possible d'utiliser plusieurs JVM, mais chacun requiert son propre gestionnaire de files d'attente, doté chacun d'un nom unique.

Lorsque MQSeries Everyplace est configuré en tant que serveur, MQSeries Everyplace est lui-même l'application initiatrice. MQSeries Everyplace prend en charge une liste de classes préchargées, lesquelles sont chargées successivement, avant que le gestionnaire de files d'attente ne le soit lui-même.

Chapitre 6. Réseaux MQSeries Everyplace et MQSeries

Bien qu'un réseau MQSeries Everyplace puisse exister de façon autonome, sans serveur ni réseau MQSeries, dans la pratique, MQSeries Everyplace est souvent utilisé pour compléter une installation MQSeries existante, étendant sa portée à de nouvelles plateformes et unités, ou fournissant des fonctions avancées, telles que des fonctions de sécurité au niveau message ou file d'attente, ou des fonctions de messagerie synchrone. Du point de vue d'une application MQSeries Everyplace, les files d'attente et les gestionnaires de files d'attente MQSeries peuvent être considérés comme de simples files d'attente et gestionnaires éloignés supplémentaires. Il existe toutefois un certain nombre de restrictions fonctionnelles, car l'accès à ces files d'attente ne s'effectue pas directement via des canaux dynamiques MQSeries Everyplace et un gestionnaire de files d'attente MQSeries Everyplace, mais requiert une passerelle MQSeries Everyplace. La passerelle peut envoyer des messages à plusieurs gestionnaires de files d'attente MQSeries, directement ou indirectement, via les canaux client MQSeries. Si la connexion est indirecte, les messages sont transmis via les canaux client MQSeries à un gestionnaire de files d'attente MQSeries intermédiaire, puis, via les canaux de message MQSeries, au gestionnaire de files d'attente cible.

Les messages issus d'une application MQSeries et destinés à MQSeries Everyplace sont adressés au gestionnaire de files d'attente MQSeries Everyplace et placés normalement dans la file d'attente, le routage MQSeries (définitions du gestionnaire de files d'attente éloigné) étant défini de sorte que les messages MQSeries Everyplace parviennent dans des files d'attente de transmission MQSeries spécifiques. Les canaux MQSeries ne sont pas définis pour les files d'attente de transmission, comme d'habitude : la passerelle MQSeries Everyplace extrait les messages de ces files d'attente et s'assure de leur livraison à leur destinataire MQSeries Everyplace. Le nombre de files d'attente de transmission à utiliser (c'est-à-dire le nombre de routes) est configurable et doit être défini en fonction du volume de messages délivrés.

Interface avec MQSeries

L'architecture de MQSeries Everyplace prend en charge le concept d'un ou de plusieurs ponts en option entre MQSeries Everyplace et les autres systèmes de messagerie.

Dans MQSeries Everyplace Version 1.0, seul un pont de ce type est admis, le pont MQSeries qui assure les interfaces entre les réseaux MQSeries Everyplace et MQSeries. Ce pont se sert du client Java MQSeries pour établir une interface avec un ou plusieurs gestionnaires de files d'attente MQSeries, permettant ainsi aux messages de transiter de MQSeries Everyplace à MQSeries et vice versa. Dans MQSeries Everyplace Version 1, un seul pont de ce type par passerelle est conseillé, chacun étant associé à plusieurs proxy de gestionnaire de files d'attente MQSeries (définitions de gestionnaires de files d'attente MQSeries). Une définition de proxy de gestionnaire de files d'attente est requise pour chaque gestionnaire MQSeries communiquant avec MQSeries Everyplace. A chacune de ces définitions peuvent être associés un ou plusieurs services de connexion client, chacun représentant une connexion à un seul gestionnaire de files d'attente MQSeries. Chacun de ces gestionnaires peut utiliser une

connexion distincte du serveur MQSeries distincte vers le gestionnaire de files d'attente et, éventuellement, un ensemble différent de propriétés, telles que les sorties utilisateur et les ports.

Un service de connexion client par passerelle peut être associé à un certain nombre de programmes d'écoute, qui passent par les services de passerelle pour se connecter au gestionnaire de files d'attente MQSeries et pour extraire les messages de MQSeries vers MQSeries Everyplace. Un programme d'écoute n'utilise qu'un service pour établir la connexion, chaque programme d'écoute étant connecté à une seule file d'attente de transmission sur le gestionnaire de files d'attente MQSeries. Chaque programme d'écoute déplace les messages d'une seule file d'attente de transmission MQSeries vers un emplacement quelconque sur le réseau MQSeries Everyplace, via son gestionnaire de files d'attente de passerelle. Ainsi un seul gestionnaire de files d'attente de passerelle peut canaliser plusieurs sources de messages MQSeries dans le réseau MQSeries Everyplace.

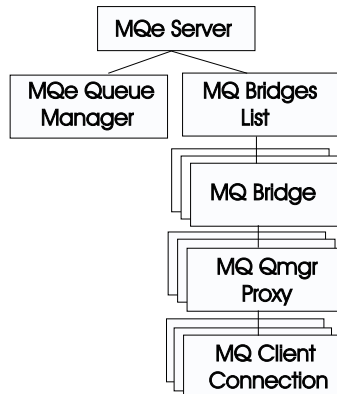


Figure 7. Hiérarchie des objets pont MQSeries

Lors de la transmission des messages dans l'autre direction, de MQSeries Everyplace vers MQSeries, le gestionnaire de files d'attente de passerelle configure un ou plusieurs objets file d'attente de pont. Chaque objet file d'attente de pont peut être connecté directement à n'importe quel gestionnaire de files d'attente et envoyer ses messages vers la file d'attente cible. De la sorte, une passerelle peut distribuer les messages MQSeries Everyplace routés via un seul gestionnaire de files d'attente MQSeries Everyplace vers n'importe quel gestionnaire de files d'attente MQSeries, directement ou indirectement. Les propriétés de l'objet pont sont indiquées dans le tableau 16.

Tableau 16. Propriétés de l'objet pont

Propriété	Explication
Nom pont	Liste de ponts
Etat exécution	Etat : actif ou arrêté

L'objet pont, de même que les autres objet passerelle, peut être démarré et arrêté indépendamment du gestionnaire de files d'attente MQSeries Everyplace. Si un objet pont de ce type est démarré (ou arrêté), l'action s'applique également à tous ses enfants (tous les ponts, les proxy de gestionnaire de files d'attente, les connexions client et les programmes d'écoute de file d'attente de transmission). Les propriétés de l'objet pont sont indiquées dans le tableau 17.

Tableau 17. Propriétés du pont

Propriété	Explication
Classe	Classe de pont
Fonction Transformation par défaut	Classe par défaut (classe de règles) à utiliser pour transformer un message de MQSeries Everyplace à MQSeries (ou vice versa) si aucune autre classe de fonction Transformation n'a été associée à la file d'attente de destination.
Intervalle des pulsations	Unité de fréquence de base à utiliser pour les opérations sur les objets pont.
Nom	Nom de l'objet pont.
Etat exécution	Etat : actif ou arrêté
Classe de règle de démarrage	Classe de règles utilisée au démarrage de l'objet pont.
Enfants du proxy du gestionnaire de files d'attente MQSeries	Liste de tous les proxy de gestionnaire de files d'attente appartenant à ce pont.

Dans les cas simples, une fonction Transformation par défaut (règle) peut être utilisée pour gérer toutes les conversions de messages. En outre, il est possible de définir une fonction Transformation par programme d'écoute (pour les messages transitant de MQSeries vers MQSeries Everyplace) qui prime sur l'option par défaut. Pour des contrôles plus spécifiques, les règles de transformation peuvent être définies sur la base de la file d'attente cible en utilisant les définitions de file d'attente de pont sur la passerelle. Cela s'applique aux files d'attente cible MQSeries Everyplace et MQSeries.

Le proxy du gestionnaire de files d'attente MQSeries contient les propriétés spécifiques d'un seul gestionnaire de files d'attente MQSeries. Les propriétés du proxy sont indiquées dans le tableau 18.

Tableau 18. Propriétés du proxy du gestionnaire de files d'attente MQSeries

Propriété	Explication
Classe	Classe du proxy du gestionnaire de files d'attente MQSeries
Nom hôte MQSeries	Nom hôte IP utilisé pour créer des connexions avec le gestionnaire de files d'attente MQSeries via les classes client Java. A défaut, le gestionnaire de files d'attente MQSeries est supposé se trouver sur la même machine que le pont et les liaisons Java sont utilisées.
Nom du proxy du gestionnaire de files d'attente MQSeries	Nom du gestionnaire de files d'attente MQSeries

Tableau 18. Propriétés du proxy du gestionnaire de files d'attente MQSeries (suite)

Propriété	Explication
Nom du pont propriétaire	Nom de l'objet pont à qui appartient le proxy du gestionnaire de files d'attente MQSeries.
Etat exécution	Etat : actif ou arrêté
Classe de règle de démarrage	Classe de règles utilisée au démarrage de l'objet gestionnaire de files d'attente MQSeries.
Enfants de la connexion client	Liste de tous les objets de connexion client qui appartiennent à ce proxy.

La définition du service de connexion client par pont contient le détail des informations requises pour établir une connexion avec un gestionnaire de files d'attente MQSeries. Les propriétés de connexion sont indiquées dans le tableau 19.

Tableau 19. Propriétés du service de connexion client

Propriété	Explication
Classe d'adaptateur	Classe à utiliser comme adaptateur de passerelle
CCSID*	Valeur entière du CCSID MQSeries à utiliser
Classe	Classe du service de connexion client par pont
Délai d'inactivité de connexion	Délai maximal d'inactivité de la connexion au-delà duquel elle est arrêtée
Mot de passe MQSeries*	Mot de passe à utiliser par le client Java
Port MQSeries*	Numéro de port IP à utiliser pour créer des connexions vers le gestionnaire de files d'attente MQSeries via les classes client Java. A défaut, le gestionnaire de files d'attente MQSeries est supposé se trouver sur la même machine que le pont et les liaisons Java sont utilisées.
Classe exit de réception MQSeries*	Utilisée pour assurer la correspondance avec l'exit de réception utilisé à l'autre extrémité du canal client ; à l'exit est associée une chaîne qui permet de transmettre les données au code de sortie.
Classe exit de sécurité MQSeries*	Utilisée pour assurer la correspondance avec l'exit de sécurité utilisé à l'autre extrémité du canal client ; à l'exit est associée une chaîne qui permet de transmettre les données au code de sortie.
Classe exit d'émission MQSeries*	Utilisée pour assurer la correspondance avec l'exit d'émission utilisé à l'autre extrémité du canal client ; à l'exit est associée une chaîne qui permet de transmettre les données au code de sortie.
ID utilisateur MQSeries*	ID utilisateur utilisé par le client Java
Nom du service de connexion client	Nom du canal de connexion du serveur sur la machine MQSeries.
Nom du proxy propriétaire du gestionnaire de files d'attente	Nom du proxy propriétaire du gestionnaire de files d'attente
Classe de règle de démarrage	Classe de règles utilisée au démarrage de l'objet service de connexion client par pont.

Tableau 19. Propriétés du service de connexion client (suite)

Propriété	Explication
Nom de file d'attente synchrone	Nom de la file d'attente MQSeries utilisée par le pont à des fins de synchronisation
Classe de règles du programme de purge de file d'attente synchrone	Classe de règles à utiliser lorsqu'un message est détecté dans la file d'attente synchrone
Etat exécution	Etat : actif ou arrêté
Nom du pont propriétaire	Nom de l'objet pont à qui appartient cette connexion client
Enfants du programme d'écoute MQ XmitQ	Liste de tous les objets programme d'écoute qui utilisent cette connexion client.
*Ces paramètres sont décrits en détail dans la documentation MQSeries - Utilisation de Java.	

La classe d'adaptateur est utilisée pour envoyer des messages de MQSeries Everyplace à MQSeries, et la file d'attente synchrone, pour garder trace de l'état de ce processus. Son contenu est utilisé dans les situations de reprise pour garantir le fonctionnement sûr de la messagerie ; après un arrêt normal, la file d'attente est vide. Elle peut être partagée par plusieurs connexions client et plusieurs définition de pont, sous réserve que les exit de réception, d'envoi et de sécurité soient identiques. Cette file d'attente peut également servir au stockage de l'état des messages transitant entre MQSeries et MQSeries Everyplace, selon les propriétés du programme d'écoute utilisé. La classe de règles du programme de purge de file d'attente synchrone est utilisée lorsqu'un message est détecté dans la file d'attente synchrone, indiquant l'échec de MQSeries Everyplace à confirmer un message.

Le délai maximal d'inactivité permet de contrôler le pool de connexions client Java, gérées par le service de connexion client de pont sur son système MQSeries. Lorsqu'une connexion MQSeries devient inactive, parce que non utilisée, un chronomètre est démarré et la connexion est abandonnée si le délai imparti expire avant que la connexion soit réutilisée. La création de connexions MQSeries est une opération coûteuse et ce processus garantit la réutilisation efficace des connexions sans consommation excessive de ressources. La valeur zéro indique qu'aucun pool de connexion ne doit être utilisé.

Les propriétés de l'objet programme d'écoute, qui transmet les messages de MQSeries vers MQSeries Everyplace, sont indiquées dans le tableau 20.

Tableau 20. Propriétés du programme d'écoute

Propriété	Explication
Classe	Classe du programme d'écoute
Nom de la file d'attente de rebut	File d'attente utilisée pour conserver les messages de MQSeries vers MQSeries Everyplace qui ne peuvent être délivrés
Adaptateur de stockage de l'état du programme d'écoute	Nom de la classe de l'adaptateur utilisé pour stocker les informations d'état.

Tableau 20. Propriétés du programme d'écoute (suite)

Propriété	Explication
Nom du programme d'écoute	Nom de la file d'attente XMIT MQSeries fournissant les messages
Nom du service de connexion client propriétaire	Nom du service de connexion client
Etat exécution	Etat : actif ou arrêté
Classe de règle de démarrage	Classe de règles utilisée au démarrage de l'objet programme d'écoute.
Classe de la fonction Transformation	Classe de règles utilisée pour déterminer la conversion d'un message MQSeries en MQSeries Everyplace
Classe de règle de message non livré	Classe de règle utilisée pour déterminer l'action déclenchée lorsque des messages de MQSeries vers MQSeries Everyplace ne peuvent pas être livrés.
Délai d'attente d'un message (en secondes)	Option avancée qui peut servir à contrôler les performances du programme d'écoute dans des circonstances exceptionnelles.

La classe de règle de message non livré détermine l'action effectuée lorsqu'un message de MQSeries vers MQSeries Everyplace ne peut pas être livré. Il est généralement placé dans la file d'attente de rebut du système MQSeries.

Pour assurer la livraison des messages, la classe du programme d'écoute utilise l'adaptateur de stockage d'état du programme d'écoute pour conserver les informations d'état, sur le système MQSeries Everyplace ou dans la file d'attente synchrone du système MQSeries.

Pour terminer la configuration du pont, les définitions du gestionnaire de files d'attente éloigné et de la file d'attente éloignée sont nécessaires. Les définitions du gestionnaire de files d'attente éloigné pour MQSeries Everyplace suivent les consignes standard MQSeries Everyplace ; dans les définitions des gestionnaires de files d'attente éloignés MQSeries, la définition du canal a la valeur NULL, pour indiquer la non utilisation d'un canal dynamique MQSeries Everyplace normal - remplacé par une connexion définie pour le gestionnaire de files d'attente MQSeries comme décrit plus haut.

La définition d'une file d'attente éloignée MQSeries Everyplace suit là aussi les consignes standard ; toutefois, pour une file d'attente MQSeries, le changement par rapport aux files d'attente MQSeries Everyplace est significatif. Les propriétés des files d'attente éloignées MQSeries sont indiquées dans le tableau 21.

Tableau 21. Propriétés des files d'attente éloignées MQSeries

Propriété	Explication
Noms d'alias	Autres noms pour la file d'attente
Authentificateur	Doit être NULL
Classe	Classe d'objet

Tableau 21. Propriétés des files d'attente éloignées MQSeries (suite)

Propriété	Explication
Connexion client	Nom du service de connexion client à utiliser
Module de compression	Doit être NULL
Module de chiffrement	Doit être NULL
Expiration	Transmise à la fonction Transformation
Taille maximale de message	Transmise à la classe de règles
Mode	Doit être synchrone
Proxy du gestionnaire de files d'attente MQ	Nom du gestionnaire de files d'attente MQSeries auquel le message doit être envoyé en premier
Pont MQSeries	Nom du pont par lequel transitera le message vers MQSeries
Nom	Nom qui identifie la file d'attente MQSeries éloignée pour MQSeries Everyplace
Gestionnaire de files d'attente propriétaire	Gestionnaire de files d'attente propriétaire de la définition
Priorité	Priorité à utiliser pour les messages (sauf si une valeur est spécifiée pour le message)
Nom de la file d'attente MQSeries éloignée	Nom de la file d'attente MQSeries éloignée
Règle	Classe de règle pour les opérations sur la file d'attente
Cible du gestionnaire de files d'attente	Gestionnaire de files d'attente MQSeries propriétaire de la file d'attente
Fonction Transformation	Nom de la classe de la fonction Transformation qui convertit le message du format MQSeries Everyplace au format MQSeries
Type	File d'attente de pont MQSeries

Les classes de l'authentificateur et des modules de chiffrement et de compression définissent un ensemble d'attributs de file d'attente qui dictent le niveau de sécurité de chaque message transmis à cette file d'attente. De l'instant, sur MQSeries Everyplace, où est envoyé initialement le message, jusqu'au moment où il est transmis à la file d'attente de pont MQSeries, le message est protégé avec un niveau de sécurité au moins égal à celui de la file d'attente. Ces niveaux de sécurité ne sont pas applicables lorsque la file d'attente du pont MQSeries transmet le message au système MQSeries. Ce transfert est effectué sous la protection des exit d'émission et de réception sur la connexion client. Aucun contrôle ne vérifie que le niveau de sécurité de la file d'attente est géré.

Les files d'attente de pont MQSeries sont exclusivement synchrones ; les applications asynchrones doivent donc envoyer les messages à ces files d'attente via les files d'attente de stockage et transmission/serveur local MQSeries Everyplace, ou via des définitions de files d'attente éloignées asynchrones.

L'administration de la passerelle est assurée de la même manière que l'administration d'un gestionnaire de files d'attente MQSeries Everyplace normal - via les messages

d'administration. De nouvelles classes de messages sont définies comme le veut l'objet géré. Le tableau 11 à la page 26 recense les classes de message d'administration de passerelle.

Conversion de message

Les messages MQSeries Everyplace destinés à MQSeries transitent par le pont et sont convertis dans un format MQSeries, via une fonction Transformation par défaut ou spécifique de la file d'attente cible. Une fonction Transformation personnalisée offre davantage de souplesse. Il est judicieux par exemple d'utiliser une sous-classe de la classe d'objet message MQSeries Everyplace pour représenter les messages d'un type particulier sur le réseau MQSeries Everyplace. Au niveau de la passerelle, une fonction Transformation peut convertir le message dans un format MQSeries par le biais de toute correspondance appropriée entre les zones et les valeurs MQSeries, ou ajouter des données spécifiques représentant la signification de la sous-classe.

La fonction Transformation MQSeries Everyplace vers MQSeries par défaut ne peut exploiter les informations de la sous-classe, mais il a été conçu pour être utilisable dans un maximum de situations. Ses caractéristiques sont les suivantes :

- **Flux des messages de MQSeries Everyplace vers MQS :**

La fonction Transformation MQSeries Everyplace vers MQSeries par défaut fonctionne conjointement avec la classe MQeMQMMsgObject. Cette classe est une représentation de toutes les zones susceptibles de figurer dans un en-tête de message MQ. A l'aide de MQeMQMMsgObject, votre application peut définir des valeurs (priorité, par exemple) via les méthodes set(). Ainsi lorsqu'un MQeMQMMsgObject (ou un objet dérivé de la classe MQeMQMMsgObject) est transmis via la fonction Transformation MQSeries Everyplace par défaut, celui-ci (MQeBaseTransformer) récupère les valeurs à l'intérieur de MQeMQMMsgObject et définit les valeurs correspondantes dans le message MQSeries (par exemple, la valeur de la priorité est copiée dans le message MQSeries).

Si le message transmis n'est ni un MQeMQMMsgObject, ni dérivé de la classe MQeMQMMsgObject, l'intégralité du message MQSeries Everyplace est copié dans le corps du message MQSeries (canalisé). La zone de format du message dans l'en-tête du message MQSeries est définie de façon à indiquer que le message MQSeries contient un message au format "canalisé" MQSeries Everyplace.

- **Flux des messages de MQSeries vers MQSeries Everyplace :**

Les messages de MQSeries vers MQSeries Everyplace et les messages transitant dans l'autre sens sont traités de façon similaire. La fonction Transformation par défaut inspecte la zone du type de message de l'en-tête MQSeries et réagit en conséquence.

Si l'en-tête MQSeries indique un message MQSeries Everyplace "canalisé", le corps du message MQSeries est reconstitué à l'identique du message MQSeries Everyplace d'origine, puis posté dans le réseau MQSeries Everyplace.

Si le message n'est pas un message MQSeries Everyplace "canalisé", le contenu de l'en-tête du message MQSeries est extrait et placé dans un objet MQeMQMMsgObject. Le corps du message MQSeries est traité comme une simple zone d'octets, et est également placé dans l'objet MQeMQMMsgObject. Le MQeMQMMsgObject est ensuite posté vers le réseau MQSeries Everyplace.

Cette classe MQeMQMsgObject et le comportement de la fonction Transformation par défaut signifient que :

- Un message MQSeries Everyplace peut transiter sans changement d'un réseau MQSeries à un réseau MQSeries Everyplace.
- Un message MQSeries peut transiter sans changement d'un réseau MQSeries Everyplace à un réseau MQSeries.
- Une application MQSeries Everyplace peut piloter toute application MQSeries existante sans la modifier.

Fonction

Au niveau de la messagerie MQSeries Everyplace, les opérations de placement (put) en mode synchrone à partir d'un gestionnaire de files d'attente MQSeries Everyplace sont activées pour les files d'attente MQSeries éloignées. Toutes les autres opérations de messagerie doivent être effectuées en mode asynchrone.

Les messages d'administration MQSeries Everyplace ne peuvent pas être envoyés à un gestionnaire de files d'attente MQSeries. Dans MQSeries Everyplace, AdminQ n'existe pas et le format des messages d'administration est différent de celui utilisé par MQSeries.

Compatibilité

Un réseau MQSeries Everyplace peut exister indépendamment de MQSeries, mais dans de nombreux cas, les deux produits sont nécessaires pour répondre aux conditions de l'application. MQSeries Everyplace peut intégrer un réseau MQSeries préexistant sous réserve de compatibilité des éléments suivants :

- **Adresse et nom** :
 - Sémantique d'adressage identique utilisant l'adresse d'un gestionnaire de files d'attente/file d'attente
 - Utilisation commune d'un espace nom ASCII.
- **Applications** : MQSeries Everyplace peut prendre en charge les applications MQSeries existantes sans changement.
- **Canaux** : Les passerelles MQSeries Everyplace utilisent les canaux client MQSeries.
- **Echange et contenu des messages** :
 - Echange de messages entre MQSeries Everyplace et MQS
 - Transparence réseau des messages (les messages issus de MQSeries Everyplace ou de MQSeries peuvent transiter par l'autre réseau sans changement).
 - Support mutuel des zones identifiées dans l'en-tête des messages MQSeries.
 - Livraison garantie unique des messages

- **Applications exemple** : Interopérabilité des applications Postcard MQSeries et du Postcard MQSeries Everyplace. ¹¹

MQSeries Everyplace Version 1 ne prend pas en charge toutes les fonctions de MQSeries. Outre les remarques concernant l'environnement, le système d'exploitation et les communications, quelques différences non négligeables sont explicitées ci-après. Notez toutefois qu'au sein de MQSeries Everyplace, nombre de tâches de l'application peuvent être exécutées par d'autres moyens utilisant les fonctions de MQSeries Everyplace, ou via l'exploitation de la transformation en sous-classes, le remplacement des classes fournies ou l'exploitation des règles, interfaces et autres fonctions de personnalisation intégrées au produit.

- Pas de support de cluster.
- Pas de support de liste de distribution.
- Pas de messages groupés/segmentés.
- Pas de fonctions d'équilibrage de charge/option de secours.
- Pas de message de référence.
- Pas d'options de rapport.
- Pas de support de file d'attente partagée.
- Pas de déclenchement.
- Pas de support d'unité de travail, pas de coordination XA.

Les caractéristiques d'évolutivité et les performances sont différentes.

Livraison garantie unique

MQSeries Everyplace et MQSeries garantissent tous deux la livraison des messages, mais à des niveaux différents. Lorsqu'un message transite de MQSeries Everyplace vers MQSeries, le transfert du message n'est assuré que si la combinaison `putMessage` et `confirmPutMessage` est utilisée (reportez-vous à la section «Opérations du gestionnaire de files d'attente» à la page 24). Lorsqu'un message transite de MQSeries vers MQSeries Everyplace, le transfert n'est assuré que si le message est défini comme persistant.

11. Cette application n'est pas intégrée à la Version 1.0, mais peut être téléchargée à partir du site de téléchargement de logiciels MQSeries sur le Web (<http://www-4.ibm.com/software/ts/MQSeries/>).

Chapitre 7. Applications et utilitaires

Remarque : Ces applications et utilitaires ne sont pas intégrés à la Version 1.0, mais peuvent être téléchargées à partir du site de téléchargement de logiciels MQSeries sur le Web(<http://www-4.ibm.com/software/ts/MQSeries/>).

Postcard

Postcard est une application MQSeries Everyplace qui permet de valider les opérations d'un réseau MQSeries Everyplace autonome ou les opérations inter-réseaux MQSeries Everyplace et MQSeries. Postcard est une application Java qui permet d'envoyer des messages texte à un utilisateur sur un gestionnaire de files d'attente éloigné. Elle inter-agit avec Postcard, de sorte qu'un message Postcard envoyé vers une cible message MQSeries est reçu sous forme de carte postale, et vice versa.

Il existe également une version C de l'application Postcard MQSeries Everyplace exécutable sur PalmOS et susceptible d'inter-agir avec la version Java.

MQSeries Everyplace Explorer

MQSeries Everyplace Explorer est un outil de gestion, écrit en Java, qui permet de configurer et d'explorer les gestionnaires de files d'attente et les files d'attente locales ou éloignées, ainsi que les messages. Il utilise les classes de base Microsoft pour présenter une interface utilisateur graphique standard Windows - mais, en conséquence, ne peut être exécuté sur une plateforme non Windows. Il peut toutefois être utilisé pour gérer tous les gestionnaires de files d'attente MQSeries Everyplace dans la mesure où il opère exclusivement par le biais d'émission et de réception de messages d'administration. Il présente une vue en deux volets d'un réseau MQSeries Everyplace, avec, dans le volet gauche, une arborescence des objets et, dans le volet droit, une liste des détails de l'objet.

Les fonctions de MQSeries Everyplace Explorer sont les suivantes :

- Affichage et modification des propriétés du gestionnaire de files d'attente
- Création, suppression et modification des connexions, et affichage de leurs propriétés
- Création, suppression et modification des files d'attente, et affichage de leurs propriétés et/ou de leur contenu
- Exploration et suppression des messages, affichage de leurs propriétés et inspection de leurs zones
- Envoi de messages de test
- Configuration du pont MQSeries Everyplace vers MQSeries

MQSeries Everyplace Explorer utilise généralement un gestionnaire de files d'attente déjà configuré et peut charger d'autres classes pour exécution. S'il n'existe pas de gestionnaire de files d'attente de ce type, il en crée un, dont les caractéristiques sont sélectionnées par l'utilisateur.

Plusieurs copies de l'outil peuvent être exécutées sur une seule machine, chacune exécutant son propre JVM. Cette disposition permet de simuler un réseau MQSeries Everyplace et peut servir à explorer et étudier les opérations et le réseau MQSeries Everyplace.

Chapitre 8. Interfaces de programmation

L'interface SPI (Systems Programming Interface) MQSeries Everyplace est l'interface de programmation de MQSeries Everyplace. Il existe deux implémentations possible, fonction du système d'exploitation. La version Java donne accès à toutes les fonctions MQSeries Everyplace ; l'interface C (MQSeries Everyplace Version 1.0) ne donne accès qu'à un sous-ensemble de ces fonctions. Pour plus de précisions sur les classes, les méthodes et les procédures, consultez le manuel MQSeries Everyplace Programming Reference ; des exemples de programmation MQSeries Everyplace sont données dans le manuel MQSeries Everyplace Programming Guide.

Annexe. Remarques

Ce document contient des informations sur des produits et des services proposés aux Etats-Unis. Ce document peut contenir des informations sur des produits, logiciels ou services IBM non annoncés dans d'autres pays. Pour plus de détails, consultez les documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM EMEA Director of Licensing
IBM Europe Middle-East Africa
Tour Descartes
La Défense 5
2, avenue Gambetta
92066 - Paris-La Défense CEDEX
France

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE «EN L'ETAT». IBM DECLINE TOUTE RESPONSABILITE, EXPLICITE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE VALEUR MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut modifier sans préavis les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

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

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux termes du Contrat sur les produits et services IBM, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Marques

Les termes qui suivent sont des marques d'International Business machines Corporation dans certains pays :

AIX AS/400 IBM MQSeries OS/390

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

D'autres sociétés sont propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

Glossaire

Ce glossaire porte sur des termes spécifiques de ce document et sur des termes utilisés avec une signification inhabituelle. Parfois, la définition donnée peut ne pas être la seule possible, mais elle indique le sens utilisé dans le document.

Si vous ne trouvez pas le terme que vous recherchez, consultez l'index ou reportez-vous à l'ouvrage IBM Dictionary of Computing, New York: McGraw-Hill, 1994.

assistant PDA (Personal Digital Assistant) : Ordinateur de poche.

authentificateur : Programme qui vérifie l'identité des expéditeurs et des destinataires des messages.

canal : Voir canal dynamique et canal MQI.

canal dynamique : Un canal dynamique connecte les unités MQSeries Everyplace et transfère les messages synchrones et asynchrones en mode bidirectionnel.

canal MQI : Un canal MQI connecte un client MQSeries à un gestionnaire de files d'attente sur un système serveur et transfère les appels et les réponses MQI en mode bidirectionnel.

classe : Collection encapsulée de données et de méthodes opérant sur les données. Une classe peut être instanciée, générant alors un objet qui est une instance de la classe.

client : Dans MQSeries, un client est un composant exécutable qui donne, aux applications utilisateur locales, accès aux services de mise en file d'attente sur un serveur.

encapsulation : Technique de programmation orientée objet qui privatise ou protège les données d'un objet de sorte que les programmeurs ne puissent y accéder et les manipuler que par le biais d'appels de méthodes.

file d'attente : Une file d'attente est un objet MQSeries. Les applications qui mettent des

messages en file d'attente peuvent y placer et y récupérer des messages.

file d'attente des messages : Voir File d'attente

gestionnaire de canal : Objet MQSeries Everyplace qui prend en charge plusieurs canaux de communication logiques concurrents entre points terminaux.

gestionnaire de files d'attente : Programme système qui fournit aux applications des services de mise en file d'attente des messages.

instance : Une instance est un objet. Lorsqu'une classe est instanciée pour générer un objet, on dit que l'objet est une instance de la classe.

interface : Classe ne contenant que des méthodes abstraites, sans aucune variable d'instance. Une interface fournit un ensemble de méthodes communes susceptibles d'être mises en oeuvre par les sous-classes de plusieurs classes différentes.

interface de programmation (API) : Une API (Application Programming Interface) est constituée des fonctions et des variables que les programmeurs sont autorisés à utiliser dans leurs applications.

Internet : Internet est un réseau public coopératif d'informations partagées. Physiquement, Internet utilise un sous-ensemble des ressources de tous les réseaux de télécommunication publics existants. Techniquement, c'est son exploitation d'un jeu de protocoles, TCP/IP (Transport Control Protocol/Internet Protocol), qui définit Internet comme réseau public coopératif.

kit JDK (Java Developers Kit) : Package logiciel destiné aux développeurs, distribué par Sun Microsystems. Il contient l'interpréteur Java, les classes et les outils de développement Java : compilateur, débogueur, désassembleur, afficheur d'applets, générateur de fichier raccord et générateur de documentation.

langage HTML (Hypertext Markup Language) :

Langage utilisé pour définir les informations à afficher sur le Web.

message : Dans les applications de mise en file d'attente de messages, un message est une communication envoyée d'un programme à un autre.

messagerie asynchrone : Méthode de communication entre programmes dans laquelle les programmes placent les messages dans des files d'attente. En mode messagerie asynchrone, le programme expéditeur poursuit les traitements en cours sans attendre la réponse à son message. S'oppose à messagerie synchrone.

messagerie synchrone : Méthode de communication entre programmes dans laquelle les programmes placent les messages dans des files d'attente. En mode messagerie synchrone, le programme expéditeur attend la réponse à son message avant de poursuivre les traitements en cours. S'oppose à messagerie asynchrone.

méthode : Terme utilisé en programmation orientée objet pour désigner une fonction ou une procédure.

mise en file d'attente de messages : Technique de programmation dans laquelle chaque programme d'une application communique avec les autres programmes en plaçant les messages en files d'attente.

module de chiffrement : Programme qui chiffre un message pour sécuriser sa transmission.

module de compression : Programme qui compresse un message pour réduire le volume des données à transmettre.

MQSeries : MQSeries est une famille de logiciels sous licence IBM qui fournissent des services de mise en file d'attente de messages.

navigateur Web : Programme qui formate et affiche des informations diffusées sur le Web.

objet : (1) En Java, un objet est une instance d'une classe. Une classe modélise un groupe d'objets ; un objet modélise un membre

spécifique de ce groupe. (2) Dans MQSeries, un objet est un gestionnaire de files d'attente, une file d'attente ou un canal.

package : En Java, un package est un moyen de donner à un module de code Java accès à un ensemble spécifique de classes. Le code Java qui fait partir d'un package a accès à toutes les classes du package et à toutes les méthodes et les zones non privées de ces classes.

passerelle : Une passerelle (ou serveur) MQSeries Everyplace est un ordinateur exécutant du code MQSeries Everyplace comportant un gestionnaire de canal.

pont : Objet MQSeries Everyplace qui permet aux messages de transiter entre MQSeries Everyplace et d'autres systèmes de messagerie, MQSeries compris.

privé : Une zone privée n'est pas visible en dehors de sa propre classe.

protégé : Une zone protégée n'est visible qu'au sein de sa propre classe, d'une sous-classe et des packages qui contiennent la classe.

protocole LDAP (Lightweight Directory Access Protocol) : Protocole client-serveur pour l'accès à un service de répertoire.

Protocole TCP/IP (Transmission Control Protocol/Internet Protocol) : Ensemble de protocoles de communication prenant en charge les fonctions de connectivité d'égal à égal sur les réseaux locaux (LAN) et longue distance (WAN).

public : Une classe ou une interface publique est visible partout. Une méthode ou une variable publique est visible partout où sa classe l'est.

serveur : (1) Un serveur MQSeries Everyplace est une unité sur laquelle est configuré un gestionnaire de canal MQSeries Everyplace. (2) Un serveur MQSeries est un gestionnaire de files d'attente qui fournit des services de mise en file d'attente de messages aux applications client exécutées sur un poste de travail éloigné. (3) Plus généralement, un serveur est un programme qui répond aux demandes d'informations, dans le modèle spécifique client/serveur de flux

d'informations entre deux programmes. (3)
Ordinateur sur lequel est exécuté un programme
serveur.

Service JNDI (Java Naming and Directory Service) : API spécifiée en langage de programmation Java. Elle fournit des fonctions d'annuaire et de répertoire aux applications écrites dans le langage de programmation Java.

servlet : Programme Java conçu pour être exécuté exclusivement sur un serveur Web.

sous-classe : Classe en étendant une autre. La sous-classe hérite des méthodes publiques et protégées et des variables de sa superclasse.

superclasse : Classe étendue par une autre. Les méthodes publiques et protégées et les variables de la superclasse sont accessibles à la sous-classe.

Web : Voir World Wide Web.

World Wide Web (Web) : Le Web est un service d'Internet, basé sur un ensemble commun de protocoles, qui permet à un serveur spécialement configuré de diffuser des documents sur Internet selon une méthode standard.

Bibliographie

Documentation connexe :

- MQSeries Everyplace - Lisez-moi d'abord, GC34-5862-00
- MQSeries Everyplace Programming Reference, SC34-5846-00
- MQSeries Everyplace Programming Guide, SC34-5845-00
- Introduction à MQSeries, GC11-1091-00
- MQSeries pour Windows NT - Mise en route - Version 5.1, GC11-1509-00

Index

A

adaptateur MQSeries Everyplace 30
administration, messages 26
administration avec MQSeries Everyplace 26
application 57
application, chargement 45
application MQSeries Everyplace 11, 57
application Postcard 57

C

canal client 7, 13
canal dynamique 7, 13, 29
capacité requise 11
chargement des applications 45
chiffrement 34
classe MQSeries Everyplace 45
client, exigences 12
client MQSeries 6
communication 43
compatibilité avec MQSeries 55
compression 34
concepts du produit 13
configuration 42
configuration MQSeries Everyplace 31
connaissance requise ix
connexion client-serveur 44
connexion commutée, gestion 30
connexion d'égal à égal 44
connexion multiple 45
conversion de message 54

D

description 1

E

enregistrement automatique 40
entité authentifiable 39
évolutivité 31
exigences des clients 12

F

famille MQSeries 5
file d'attente, pont MQSeries 20
file d'attente, stockage et réacheminement 19
file d'attente du serveur local 19
file d'attente éloignée 19
file d'attente locale 18
file d'attente MQSeries Everyplace 18
fonction Transformation 54
format des données de vidage 18

G

gestionnaire de canal 43
gestionnaire de files d'attente 6, 24, 43

gestionnaire de files d'attente MQSeries Everyplace 22

I

interface avec MQSeries 47
interface de programmation 59
interface de sécurité 41

J

journal des événements 31

L

lecteur du manuel ix
légale, remarque 61
livraison garantie unique des messages 56

M

marque déposée 62
message, livraison garantie unique 56
messagerie asynchrone 22, 33
messagerie de la station de travail 6
messagerie diffuse 6
messagerie distribuée 5
messagerie hôte 5
messagerie MQSeries 5
messagerie synchrone 23, 34
messages d'administration 26
mini-certificat 39
MQeAttribute 37
MQeMTrustAttribute 38
MQSeries, compatibilité 55
MQSeries, interface 47
MQSeries Everyplace, administration 26
MQSeries Everyplace, utilitaire d'administration 57
MQSeries Everyplace, utilitaires 57
MQSeries Workflow 5
MQSeriesIntegrator 5

O

objet message 14
objet MQSeries Everyplace 14
objet pont 49
objet programme d'écoute 51, 52
objet proxy du gestionnaire de files d'attente 49

P

passerelle MQSeries Everyplace 7, 13
personnalisation 42
pont, objet 48
pont MQS 47
pont MQSeries 47
prérequis 3
présentation ix
programme d'écoute de canal 43

R

registre 38
registre Everyplace MQSeries 13
registre MQSeries Everyplace 13, 38
registre privé 39
registre public 40
règles d'attribut 42
règles de file d'attente 43
règles du composeur RAS 42
règles du gestionnaire de files d'attente 43
règles du pont MQSeries 42
règles MQSeries Everyplace 42
remarque légale 61
réplication de certificat 40
réseau MQS 47
réseau MQSeries 47
réseau MQSeries Everyplace 31, 47

S

sécurité au niveau des files d'attente 36
sécurité au niveau message 37
sécurité locale 35
sécurité MQSeries Everyplace 34
serveur local MQSeries Everyplace 19
serveur MQSeries 6
service d'émission de mini-certificats 41
SPI 59
stockage et réacheminement, file d'attente 19
style de connexion 43
style de connexion multiple 45
surveillance 29
système d'exploitation pris en charge 3
système d'exploitation requis 3

T

terminologie ix
trace MQSeries Everyplace 31

U

unité MQSeries Everyplace 7, 13
utilitaire 57
utilitaire d'administration 57
utilitaire MQSeries Everyplace Explorer 57

IBM

GC11-1701-00

