

Composants IBM WebSphere Business Integration
Adapter



Adapter for e-Mail - Guide d'utilisation

Adapter Version 5.5.x

Composants IBM WebSphere Business Integration
Adapter



Adapter for e-Mail - Guide d'utilisation

Adapter Version 5.5.x

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant dans la section «Informations légales», à la page 101.

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 2005. Tous droits réservés.

© **Copyright International Business Machines Corporation 2000, 2005. All rights reserved.**

Table des matières

Avis aux lecteurs canadiens	v
A propos de ce document	vii
A propos de ce document	vii
Contenu du document	vii
Limites du document	vii
Public visé	viii
Documents connexes	viii
Conventions typographiques	ix
Nouveautés de la présente édition	xi
Version 5.5.x	xi
Version 5.4.x	xi
Version 5.3.x	xi
Version 5.2.x	xii
Versions précédentes	xii
Chapitre 1. Présentation de l'adaptateur	1
Fonctionnalités de l'adaptateur E-Mail	1
Composants de l'adaptateur	2
Fonctionnement de l'adaptateur	3
Exemple de scénario pour l'adaptateur pour e-mail	8
Traitement des données dépendantes des paramètres régionaux	10
Common Event Infrastructure	11
Application Response Measurement	11
Chapitre 2. Installation et configuration de l'adaptateur	13
Environnement de l'adaptateur pour e-mail	13
Détermination des tâches d'installation et de configuration	15
Installation de l'adaptateur pour e-mail et des fichiers associés	17
Vérification de l'installation	17
Tâches de configuration de l'adaptateur	18
Configuration de la définition de l'adaptateur	19
Création de plusieurs instances de connecteur	23
Démarrage du connecteur	24
Arrêt du connecteur	26
Chapitre 3. Planification de l'implémentation d'un adaptateur pour e-mail	27
Présentation de l'implémentation d'un adaptateur pour e-mail	27
Fonctionnement de l'adaptateur pour e-mail et des gestionnaires de données	28
Exemples de fichiers de l'adaptateur pour e-mail	29
Chapitre 4. Développement d'objets métier pour l'adaptateur	31
Utilisation des objets métier de l'adaptateur pour e-mail	31
Meta-objets du gestionnaire de données	39
Utilisation des méta-objets de l'adaptateur pour e-mail	40
Objet métier EmailNotification	43
Indication d'adresses e-mail valides	45
Annexe A. Propriétés standard de configuration pour les adaptateurs	47
Nouvelles propriétés	47
Présentation des propriétés de connecteur standard	47
Référence rapide des propriétés standard	49
Propriétés standard	55

Annexe B. Connector Configurator	73
Présentation de Connector Configurator	73
Démarrage de Connector Configurator	74
Exécution de Connector Configurator à partir de System Manager	75
Création d'un modèle de propriétés spécifiques au connecteur	75
Création d'un fichier de configuration	78
Utilisation d'un fichier existant	80
Remplissage d'un fichier de configuration	81
Définition des propriétés d'un fichier de configuration	81
Enregistrement de votre fichier de configuration	90
Modification d'un fichier de configuration	90
Exécution de la configuration	91
Utilisation de Connector Configurator dans un environnement globalisé	91
Annexe C. Application Response Measurement	93
Prise en charge des appels Application Response Measurement	93
Annexe D. Common Event Infrastructure	95
Logiciels requis	95
Activation de Common Event Infrastructure	95
Obtention d'événements d'adaptateur Common Event Infrastructure	95
Pour plus d'informations	96
Définitions du catalogue d'événements Common Event Infrastructure	96
Format XML des métadonnées de "start adapter"	96
Format XML des métadonnées de "stop adapter"	98
Format XML des métadonnées de "timeout adapter"	98
Format XML des métadonnées de "request" ou "delivery"	99
Informations légales	101
Informations sur les interfaces de programmation	103
Marques et marques de service	103
Index	105

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

A propos de ce document

A propos de ce document

La famille de produits IBM^R WebSphere^R Business Integration Adapter propose une connectivité d'intégration pour les technologies e-business de pointe, les applications d'entreprise, les systèmes propriétaires et les grands systèmes. Cette famille de produits fournit des outils et des modèles destinés à la personnalisation, la création et la gestion des composants pour l'intégration métier.

Contenu du document

Ce document décrit l'installation, la configuration des propriétés du connecteur, le développement d'objets métier et la résolution des incidents de IBM WebSphere Business Integration Adapter for e-Mail.

Limites du document

Ce document ne décrit pas les indicateurs de déploiement et ne traite pas les questions relatives à la planification de la capacité, telles que l'équilibrage de la charge, le nombre d'unités d'exécution de l'adaptateur, les débits maximum et minimum et les seuils de tolérance.

Ces questions dépendent du déploiement de chaque utilisateur et doivent être étudiées dans les conditions les plus proches possibles de l'environnement exact dans lequel l'adaptateur doit être déployé. Contactez le service de maintenance IBM pour discuter de la configuration de votre site de déploiement, afin d'obtenir des informations sur la planification et l'évaluation de ces types d'indicateurs, en fonction de votre configuration.

Ce document décrit l'installation, la configuration, le développement d'objets métier et la résolution des incidents de IBM WebSphere Business Integration Adapter for e-Mail.

Contenu du document

Ce document décrit l'installation, la configuration des propriétés du connecteur, le développement d'objets métier et la résolution des incidents de IBM WebSphere Business Integration Adapter for e-Mail.

Limites du document

Ce document ne décrit pas les indicateurs de déploiement et ne traite pas les questions relatives à la planification de la capacité, telles que l'équilibrage de la charge, le nombre d'unités d'exécution de l'adaptateur, les débits maximum et minimum et les seuils de tolérance.

Ces questions dépendent du déploiement de chaque utilisateur et doivent être étudiées dans les conditions les plus proches possibles de l'environnement exact dans lequel l'adaptateur doit être déployé. Contactez le service de maintenance IBM pour discuter de la configuration de votre site de déploiement, afin d'obtenir des informations sur la planification et l'évaluation de ces types d'indicateurs, en fonction de votre configuration.

Public visé

Ce document est destiné aux consultants IBM et aux clients. Les utilisateurs doivent bien connaître les systèmes d'intégration WebSphere et les technologies de messagerie électronique.

IBM WebSphere Business Integration Adapters propose une connectivité d'intégration pour les technologies e-business de pointe et les applications d'entreprise.

Ce document décrit l'installation, la configuration et le développement d'objets métier de l'adaptateur pour e-mail.

Documents connexes

La documentation complète qui accompagne ce produit présente les caractéristiques et les fonctions communes à toutes les installations de composants WebSphere Business Integration Adapter, et inclut des supports de référence sur des composants spécifiques.

Vous pouvez télécharger la documentation associée sur les sites suivants :

- Pour obtenir des informations générales sur les adaptateurs, pour apprendre à les utiliser avec des courtiers de messages WebSphere (WebSphere MQ Integrator, WebSphere MQ Integrator Broker, WebSphere Business Integration Message Broker) et avec WebSphere Application Server, voir le centre d'information IBM WebSphere Business Integration Adapters à l'adresse : <http://www.ibm.com/websphere/integration/wbiadapters/infocenter>
- Pour utiliser les adaptateurs avec WebSphere InterChange Server, voir les centres d'information IBM WebSphere InterChange Server : <http://www.ibm.com/websphere/integration/wicserver/infocenter>
<http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>
- Pour plus d'informations sur les courtiers de messages WebSphere : <http://www.ibm.com/software/integration/mqfamily/library/manualsa/>
- Pour plus d'informations sur WebSphere Application Server : <http://www.ibm.com/software/webservers/appserv/library.html>

Ces sites contiennent des explications simples pour télécharger, installer et afficher la documentation.

Remarque : Des informations importantes relatives à ce produit peuvent être disponibles dans les flashes de support technique (Technical Support Flashes), après la publication de ce document. Pour les consulter, accédez au site du support de WebSphere Business Integration, <http://www.ibm.com/software/integration/websphere/support/>. Sélectionnez la zone qui vous intéresse et parcourez les sections Technotes et Flashes.

Conventions typographiques

Ce document utilise les conventions suivantes :

Police courier	Indique une valeur littérale, comme le nom d'une commande, le nom d'un fichier, des informations que vous tapez ou que le système affiche à l'écran.
gras	Indique un nouveau terme à sa première occurrence.
<i>italique italic</i>	Indique un nom de variable ou une référence croisée.
<i>texte en bleu</i>	Un texte en bleu, visible uniquement lorsque vous consultez le document en ligne, indique un hyperlien de référence croisée. Si vous cliquez sur le texte en bleu, vous êtes renvoyé à l'objet de la référence.
{ }	Dans une ligne de syntaxe, les accolades entourent un ensemble d'options parmi lesquelles vous ne devez sélectionner qu'une seule.
[]	Dans une ligne de syntaxe, les crochets entourent un paramètre facultatif.
...	Dans une ligne de syntaxe, les points de suspension indiquent une répétition du paramètre précédent. Par exemple, <code>option[,...]</code> signifie que vous pouvez entrer plusieurs options séparées par des virgules.
< >	Dans une convention de dénomination, les signes inférieur à et supérieur à entourent les différents éléments d'un nom afin de pouvoir les différencier les uns des autres, par exemple, <code><nom_serveur><nom_connecteur>tmp.log</code> .
/, \	Dans ce document, les barres obliques inverses (\) sont utilisées comme convention pour les chemins de répertoire. Pour les systèmes UNIX, remplacez les barres obliques inverses par des barres obliques (/). Tous les noms de chemin des produits de systèmes d'intégration WebSphere sont relatifs au répertoire dans lequel le produit est installé sur votre système.
%texte% et \$texte	Du texte placé entre des signes pourcentage (%) indique la valeur de la variable système texte Windows ou la variable utilisateur. Le symbole équivalent dans un environnement UNIX est <code>\$texte</code> , indiquant la valeur de la variable d'environnement UNIX <code>texte</code> .
<i>ProductDir</i>	La famille de produits est WBIA : il s'agit du répertoire dans lequel le produit IBM WebSphere Business Integration Adapters est installé. La variable d'environnement CROSSWORLDS contient le chemin du répertoire ProductDir, par défaut IBM\WebSphereAdapters.

Nouveautés de la présente édition

Version 5.5.x

- Prise en charge d'AIX 5.3
- Capacité de paramétrer l'en-tête de l'e-mail ContentType (permet l'envoi de messages HTML)
- Prise en charge de la définition de noms de pièces jointes sortantes définis par l'utilisateur
- La structure des objets métier entrants accepte désormais les pièces jointes en tant qu'enfants
- Capacité de définir les propriétés IMAP Session lors de la connexion au serveur de messagerie
- Prise en charge de pièces jointes au format binaire
- Nouvelle propriété de méta-objet ExternalProperty

Version 5.4.x

La mise à jour de l'adaptateur pour e-mail porte sur les points suivants :

- Sous Windows, l'adaptateur prend en charge l'écriture bidirectionnelle
- Activation de Tivoli License Manager
- Prise en charge du code chinois GB18030
- Ajout de la fonctionnalité ReplyTo
- Ajout de la prise en charge de la spécification RFC822 pour la syntaxe des adresses Internet
- Pour cette édition, le code d'exécution de l'adaptateur a été séparé de la ligne de code du serveur

La version 5.4.x de l'adaptateur est prise en charge sur les plateformes suivantes :

- WIN 2003
- AIX 5.1 et 5.2 : cet adaptateur prend en charge une JVM 32 bits sur une plateforme 64 bits
- Solaris 9 : cet adaptateur prend en charge une JVM 32 bits sur une plateforme 64 bits
- HP-UX, 11, 11i
- Linux RedHat AS 3.0, ES 3.0 et WS 3.0
- SUSE LINUX Standard Server 8.1 et Enterprise Server 8.1 SP3
- IBM JRE/JDK 1.4.2

Version 5.3.x

L'adaptateur pour e-mail a reçu les correctifs de maintenance générale.

A partir de la version 5.3, l'adaptateur pour e-mail n'est plus pris en charge par la plateforme Solaris 7.

Version 5.2.x

L'adaptateur pour e-mail a reçu les correctifs de maintenance générale.

A partir de la version 5.2, l'adaptateur pour e-mail n'est plus pris en charge sous Microsoft Windows NT.

Les informations relatives à l'installation de l'adaptateur ont été déplacées. Voir le chapitre 2, «Installation de l'adaptateur pour e-mail et des fichiers associés», à la page 17, pour savoir où sont ces informations.

Versions précédentes

Nouvelles fonctionnalités et autres modifications apportées aux versions précédentes.

Version 5.1.x

L'adaptateur peut désormais utiliser WebSphere Application Server comme courtier d'intégration. Pour plus d'informations, voir «Compatibilité du courtier», à la page 13.

L'adaptateur fonctionne désormais sur les plateformes suivantes :

- Solaris 7, 8
- AIX 5.x
- HP UX 11.i

Version 5.0.x

Mise à jour de mars 2003. Le nom "CrossWorlds" n'est plus utilisé pour décrire un système complet ou pour modifier les noms des composants ou outils, qui restent par ailleurs quasiment inchangés. Par exemple, "CrossWorlds System Manager" est devenu "System Manager," et "CrossWorlds InterChange Server" est maintenant "WebSphere InterChange Server".

IBM WebSphere Business Integration Adapter for e-Mail est proposé avec les mêmes fonctionnalités que dans les éditions précédentes.

L'édition de ce document contient les nouveautés ou modifications suivantes :

- Informations sur la propriété d'adaptateur DefaultVerb, qui permet à l'adaptateur de fournir une instruction par défaut pour les objets métier générés à partir d'un fichier si celui-ci ne contient aucune information de ce type. Pour plus d'informations, voir «DefaultVerb», à la page 22.
- Informations sur l'attribut de méta-objet AttachmentExtension, qui permet à l'adaptateur de préciser une extension pour les fichiers qu'il génère, représentant l'objet métier contenu dans un message électronique. Pour plus d'informations, voir «AttachmentExtension», à la page 32.
- L'attribut de méta-objet enfant précédemment appelé EventRecovery a été renommé InDoubtEvents. Les valeurs possibles pour cet attribut sont désormais Reprocess, FailOnStartup, LogError et Ignore. Pour plus d'informations, voir «InDoubtEvents», à la page 42.

- Le nouvel attribut de méta-objet enfant Reconnect a été ajouté à `MQ_PollLocation`. Les valeurs possibles pour cet attribut sont `true` ou `false`. Ces valeurs seront utilisées pour déterminer si une tentative de reconnexion sera effectuée après perte de la connexion. Pour plus d'informations, voir «Notification d'événement», à la page 4.

Version 4.3.x

IBM WebSphere Business Integration for e-Mail inclut l'adaptateur pour e-mail. Celui-ci fonctionne avec les courtiers d'intégration InterChange Server et WebSphere MQ Integrator. Un courtier d'intégration est une application qui permet l'intégration d'ensembles hétérogènes d'applications. Il fournit divers services dont le routage des données.

Cet adaptateur comprend :

- Un composant spécifique à l'application, spécifiquement pour la messagerie électronique.
- Un exemple, situé dans `\connectors\EMail\samples`.
- IBM WebSphere Adapter Framework, composé de :
 - Adapter Framework
 - Des outils de développement (y compris Business Object Designer et Connector Configurator)
 - Des API (y compris ODK, JCDK et CDK)

Ce manuel fournit des instructions sur l'utilisation de cet adaptateur avec les courtiers d'intégration InterChange Server et WebSphere MQ Integrator.

L'adaptateur a été internationalisé. Pour plus d'informations, voir «Traitement des données dépendantes des paramètres régionaux», à la page 10 et Annexe A, «Propriétés standard de configuration pour les adaptateurs», à la page 47.

Chapitre 1. Présentation de l'adaptateur

Le présent chapitre décrit l'adaptateur pour e-mail et ses composants. L'adaptateur permet à un courtier d'intégration d'échanger des objets métier, des fichiers et des messages avec des applications de messagerie électronique. Grâce à lui, le courtier d'intégration IBM WebSphere InterChange Server peut envoyer des e-mails aux utilisateurs désignés, en cas de messages d'erreur ou d'erreur irrémédiable. Il contient les sections suivantes :

- «Fonctionnalités de l'adaptateur E-Mail»
- «Composants de l'adaptateur», à la page 2
- «Fonctionnement de l'adaptateur», à la page 3
- «Exemple de scénario pour l'adaptateur pour e-mail», à la page 8
- «Traitement des données dépendantes des paramètres régionaux», à la page 10
- «Common Event Infrastructure», à la page 11
- «Application Response Measurement», à la page 11

Fonctionnalités de l'adaptateur E-Mail

L'adaptateur pour e-mail a les fonctionnalités suivantes :

- **Traitement de l'objet métier** – L'adaptateur pour e-mail permet à un courtier d'intégration d'envoyer à des applications de messagerie des courriers électroniques contenant des objets métier ou des fichiers.

L'adaptateur peut également interroger un ou plusieurs serveurs de messagerie pour surveiller l'arrivée de messages et les convertir en objets métier qu'il envoie ensuite au courtier d'intégration. L'encapsulation d'objets métier dans des e-mails permet aux entreprises d'intégrer des processus métier et de synchroniser les données sur Internet.

- **Notification d'erreur (courtier d'intégration InterChange Server uniquement)** – L'adaptateur pour e-mail permet à IBM WebSphere InterChange Server d'utiliser l'e-mail pour envoyer des messages d'erreur. InterChange Server envoie un e-mail pour indiquer aux utilisateurs d'éventuels messages d'erreur ou d'erreur irrémédiable.

L'adaptateur pour e-mail travaille avec la collaboration de messagerie électronique, créée automatiquement par InterChange Server. Lorsqu'une erreur se produit et que la notification par e-mail est configurée, InterChange Server crée un objet métier de notification par e-mail contenant les informations sur le message, et le transmet à l'adaptateur pour e-mail. L'adaptateur extrait les informations du message depuis l'objet métier, crée le message électronique et l'envoie.

Remarque : Le traitement de l'objet métier et la notification d'erreur sont deux rôles distincts de l'adaptateur. Pour activer ces deux fonctions, vous devez les configurer. Pour plus d'informations, voir le Chapitre 2, «Installation et configuration de l'adaptateur», à la page 13.

Composants de l'adaptateur

Les adaptateurs se composent de deux parties : l'architecture de l'adaptateur et le composant spécifique à l'application. L'architecture de l'adaptateur, dont le code est commun à tous les adaptateurs, joue le rôle d'intermédiaire entre le courtier d'intégration et le composant propre à l'application. Le composant propre à l'application contient des codes adaptés à une application ou à une technologie donnée. L'architecture de l'adaptateur fournit les services suivants entre le courtier d'intégration et le composant propre à l'application :

- Elle reçoit et envoie des objets métier.
- Elle assure l'échange des messages de démarrage et d'administration.

Le présent document contient des informations sur l'architecture de l'adaptateur et le composant spécifique à l'application, désigné comme étant l'adaptateur. WebSphere Business Integration Adapter for e-Mail utilise également un gestionnaire de données, qui convertit un objet métier dans un format de données spécifique et convertit les pièces jointes au message dans un format spécifique aux objets métier. L'adaptateur peut utiliser un des gestionnaires de données fournis par IBM, ou bien un gestionnaire de données personnalisé. Pour plus d'informations sur les gestionnaires de données, voir le Chapitre 3, «Planification de l'implémentation d'un adaptateur pour e-mail», à la page 27 et le *Data Handler Guide*.

La figure 1 illustre l'architecture de l'adaptateur pour e-mail.

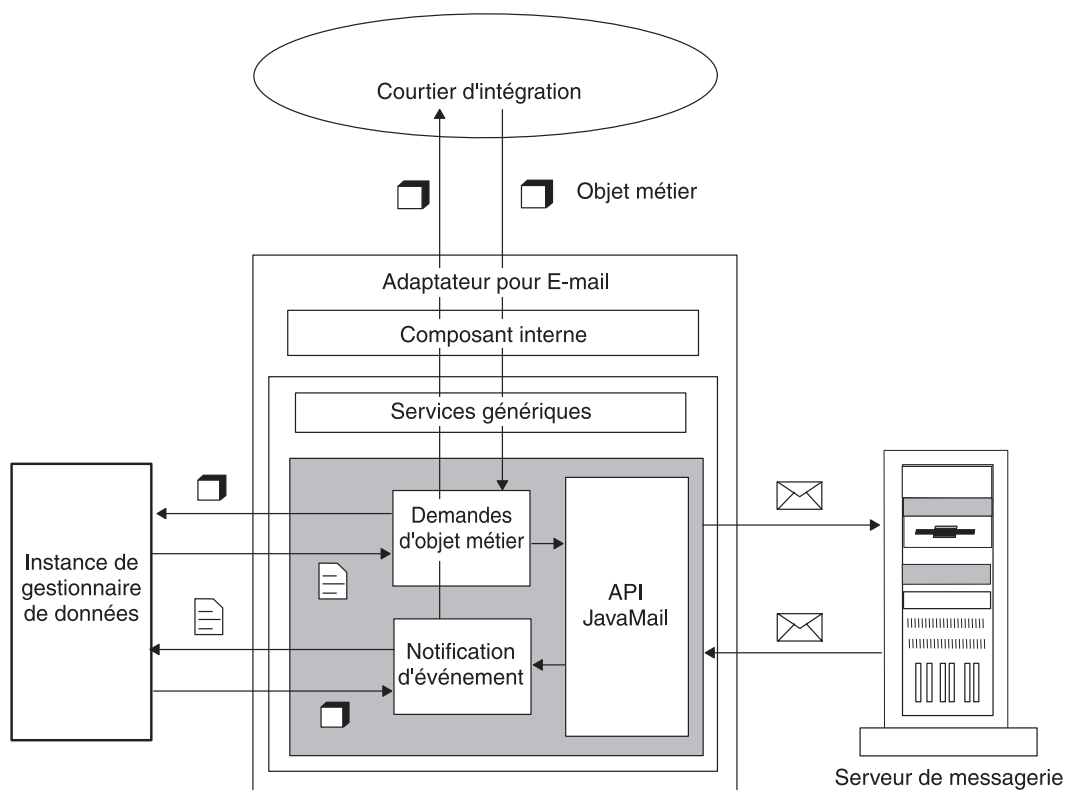


Figure 1. Architecture de l'adaptateur pour e-mail

Protocoles de transfert

L'adaptateur interagit avec les serveurs de messagerie à l'aide de JavaMail. L'adaptateur fournit les paramètres requis par JavaMail pour communiquer avec les serveurs de messagerie, et JavaMail gère les protocoles SMTP (Simple Mail Transport Protocol) et IMAP (Internet Message Access Protocol).

SMTP est le protocole de transfert de messagerie Internet. Il est utilisé par l'adaptateur pour transférer les messages.

Pour l'interrogation, l'adaptateur pour e-mail prend en charge le protocole IMAP via JavaMail pour accéder en ligne aux messages e-mail et la manipulation à distance des boîtes de réception. IMAP sert pour enregistrer et manipuler les messages sur un serveur de messagerie, autorisant ainsi l'accès aux e-mails depuis n'importe où. IMAP prend en charge les requêtes en ligne, l'extraction sélective du contenu des messages et les recherches de message au niveau du serveur. Notez qu'actuellement l'adaptateur ne prend pas en charge le protocole POP via JavaMail.

Fonctionnement de l'adaptateur

Les sections qui suivent décrivent comment l'adaptateur traite les requêtes d'objets métier et gère les événements.

Traitement des objets métier

Lorsqu'un courtier d'intégration envoie un objet métier à l'adaptateur pour e-mail, ce dernier traite l'objet métier et génère un e-mail.

Un objet métier e-mail doit contenir des informations de routage du courrier électronique (adresse de l'émetteur, adresse du destinataire et contenu), le type MIME des pièces jointes et les objets métier à livrer. L'objet métier peut également préciser le chemin complet des fichiers à joindre à l'e-mail. Un même objet métier peut aboutir à un e-mail contenant plusieurs objets métier et plusieurs fichiers en pièce jointe.

Pour traiter une demande d'objet métier, l'adaptateur :

1. Extrait les informations de routage de l'objet métier de premier niveau et compose un e-mail.
2. Traite chaque objet métier ou fichier contenu de la façon suivante :
 - En présence d'objets métier contenus, l'adaptateur appelle un gestionnaire de données pour convertir chaque objet métier en une chaîne d'objet métier, qui est fournie en pièce jointe à l'e-mail. Le type MIME de la pièce jointe est défini dans l'objet métier de premier niveau. Il y a un seul objet métier par pièce jointe.
 - Si des fichiers sont précisés dans l'objet métier de premier niveau, l'adaptateur les inclut en tant que pièces jointes à l'e-mail. Chaque fichier doit être situé à l'emplacement précisé.
3. Une fois l'e-mail terminé, l'adaptateur le livre à l'aide de JavaMail.

La figure 2 illustre le traitement de l'objet métier par l'adaptateur pour e-mail. Pour obtenir des informations sur la création d'objets métier pour l'adaptateur, voir le Chapitre 4, «Développement d'objets métier pour l'adaptateur», à la page 31.

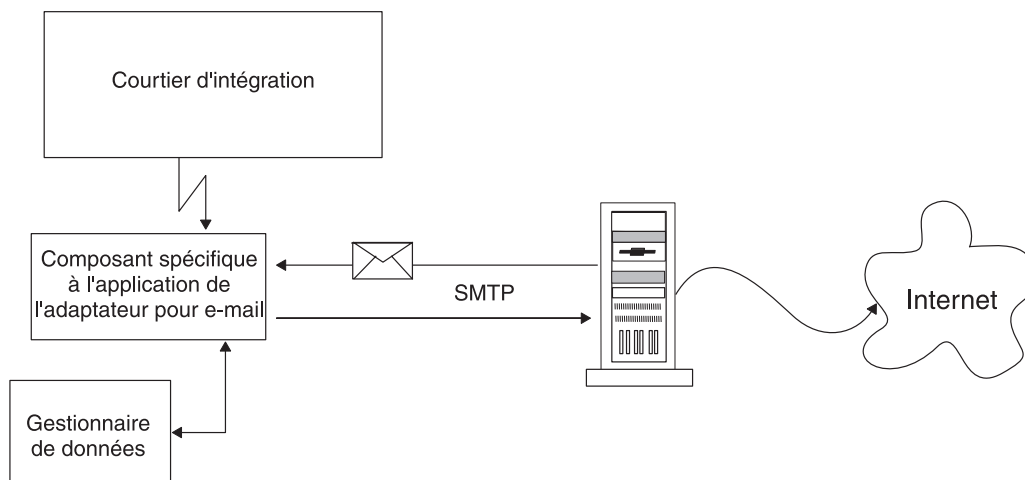


Figure 2. Traitement de l'objet métier de l'adaptateur pour e-mail

Notification d'événement

Dans la notification d'événement, l'adaptateur demande si des e-mails sont présents dans les dossiers spécifiés sur le serveur de messagerie. L'adaptateur contrôle les dossiers, extrait les nouveaux messages, les convertit en objets métier à l'aide du gestionnaire de données, et envoie les objets métier au courtier d'intégration.

L'adaptateur peut interroger plusieurs emplacements. Les dossiers suivants sont requis par emplacement d'interrogation :

- Dossier d'interrogation – Le dossier interrogé par l'adaptateur.
- Dossier d'archivage – Le dossier dans lequel l'adaptateur archive les événements réussis.
- Dossier des événements échoués – Le dossier dans lequel l'adaptateur archive les événements qui ont échoué.

Les emplacements d'interrogation peuvent se situer dans différents magasins IMAP, mais ces trois dossiers, s'ils sont indiqués pour un emplacement d'interrogation, doivent être dans la même boîte aux lettres.

Les sections qui suivent décrivent le fonctionnement des événements de façon plus détaillée.

Déclenchement d'événement

Un événement déclenchant pour un adaptateur est un nouvel e-mail qui arrive dans un dossier d'interrogation. L'adaptateur contrôle dans chaque dossier d'interrogation la présence de nouveaux e-mails, selon un intervalle défini par la propriété de configuration PollFrequency.

Détection d'événement

Chaque serveur de messagerie IMAP dispose d'un magasin de messages appelé magasin IMAP. Le magasin IMAP contient une boîte aux lettres pour chacun des utilisateurs du serveur de messagerie. Une boîte aux lettres peut se composer de plusieurs dossiers, pouvant chacun contenir plusieurs e-mails ou dossiers.

Chaque dossier d'interrogation doit disposer d'un dossier d'archivage et d'un dossier d'événements échoués correspondants. Le dossier d'interrogation, le dossier d'archivage et le dossier des événements échoués sont regroupés sous l'appellation d'**emplacement d'interrogation**. Un administrateur doit créer les dossiers et préciser les emplacements d'interrogation dans le méta-objet de l'adaptateur.

L'adaptateur recherche les nouveaux messages dans le dossier d'interrogation d'un magasin IMAP. Il extrait alors les nouveaux messages qui s'y trouvent. L'adaptateur détermine qu'un message est nouveau en vérifiant le statut de son indicateur IMAP.

Statut d'événement : IMAP prend en charge des indicateurs qui décrivent l'état des e-mails dans une boîte aux lettres. Le mécanisme de détection des événements les utilise pour identifier les messages à extraire. Le statut d'un événement est spécifié par le dossier dans lequel le message est situé et par l'indicateur du message.

Le mécanisme d'événement utilise l'indicateur IMAP SEEN pour marquer les événements qui sont "en cours". Lors de l'interrogation, l'adaptateur contrôle les messages qui figurent dans le dossier d'interrogation, et prélève ceux qui ne sont pas marqués de l'indicateur SEEN. Ces événements sont "prêts à être interrogés". Lorsque l'adaptateur prélève un message dans le dossier d'interrogation pour le traiter, il lui attribue l'indicateur SEEN. Les événements marqués de l'indicateur SEEN sont "en cours".

Remarque : L'adaptateur n'interroge pas les messages marqués de l'indicateur SEEN.

Une fois le traitement terminé et que toutes les pièces jointes d'un message ont été converties en objets métier et envoyées au courtier d'intégration, l'e-mail est marqué de l'indicateur FLAGGED et déplacé dans le dossier d'archivage. Les messages dont les pièces jointes ne peuvent être converties en objets métier, ceux qui ont des objets métier non souscrits ou dont les objets métier n'ont pas pu être livrés au courtier d'intégration, sont déplacés dans le dossier des événements échoués. Le tableau 1 résume les états des événements de l'adaptateur.

Tableau 1. Etats des événements

Etats des événements	Etat de l'e-Mail
prêt à être interrogé	Tous les messages du dossier d'interrogation et dont l'indicateur SEEN n'est pas défini.
en cours	Tous les messages du dossier d'interrogation dont l'indicateur SEEN est défini.
envoyé au courtier d'interrogation	Tous les messages du dossier d'archivage. Tout message de PollFolder marqué de l'indicateur FLAGGED. Remarque : L'adaptateur fait passer l'indicateur d'un message depuis IN_PROGRESS vers FLAGGED, une fois qu'il a été envoyé au courtier d'échange et prêt à être archivé (déplacé dans le dossier d'archivage).
objets métier non souscrits et erreurs	Tous les messages du dossier des événements échoués.

Extraction des événements

Pour extraire les événements, l'adaptateur interroge chaque dossier l'un après l'autre et prélève un message dans chaque boîte. Cette stratégie d'interrogation garantit le traitement de tous les messages de tous les dossiers.

L'adaptateur procède comme suit pour traiter chacun des e-mails extraits d'un emplacement d'interrogation :

1. Dans le dossier d'interrogation, l'adaptateur prélève un e-mail sans indicateur SEEN. Il lui attribue alors l'indicateur SEEN.
2. L'adaptateur récupère l'e-mail depuis le dossier d'interrogation. Toutes les pièces jointes sont traitées, quel que soit leur nombre. Il doit y avoir un seul objet métier par pièce jointe.
3. Pour chaque pièce jointe, l'adaptateur utilise le type MIME correspondant pour déterminer quel gestionnaire de données utiliser pour convertir la pièce jointe en objet métier. L'adaptateur extrait le contenu de la pièce jointe et le transmet au gestionnaire de données. Les pièces jointes d'un e-mail peuvent avoir des formats différents. Si la pièce jointe n'a pas de type MIME, l'adaptateur ne la traite pas.
4. Le gestionnaire de données génère un objet métier et le retourne à l'adaptateur.
5. Si l'objet métier est souscrit, l'adaptateur le transmet au courtier d'intégration. Dans le cas contraire, l'adaptateur considère cet e-mail comme étant un événement échoué.

L'adaptateur traite toutes les pièces jointes de la même façon. Un e-mail est considéré comme un événement réussi si toutes les pièces jointes ont été converties en objets métier et transmises au courtier d'intégration. En cas de réussite, l'adaptateur marque les e-mails de l'indicateur FLAGGED, puis les conserve dans le dossier d'archivage de cet emplacement d'interrogation. L'adaptateur n'archive l'e-mail qu'une fois toutes les pièces jointes traitées.

Si une ou plusieurs pièces jointes n'ont pu être converties en objets métier, ou si les objets métier fournis par le gestionnaire de données ne sont pas souscrits, l'événement est considéré comme échoué. Dans ce cas, l'adaptateur conserve l'e-mail dans le dossier des événements échoués de cet emplacement d'interrogation.

L'adaptateur fournit un nombre d'événements par dossier inférieur ou égal à la valeur indiquée par la propriété PollQuantity de l'adaptateur. L'adaptateur ne prélève et ne traite pas d'autre e-mail tant qu'il n'en a pas terminé avec le message

Événements échoués : Un événement est considéré comme ayant échoué dans les cas suivants :

- Une pièce jointe n'a pas pu être extraite.
- L'adaptateur n'a pas pu déterminer quel gestionnaire de données utiliser pour une pièce jointe.
- Le gestionnaire de données n'a pas pu convertir une pièce jointe en un objet métier.
- L'objet métier retourné par un gestionnaire de données n'est pas souscrit.

L'adaptateur traite toutes les pièces jointes d'un message et fournit les objets métier qu'il génère au courtier d'intégration, même si le message est un événement échoué. Par exemple, prenons un e-mail contenant cinq pièces jointes. L'adaptateur traite les pièces jointes un et deux, ne parvient pas à traiter la troisième et réussit à traiter les pièces jointes quatre et cinq. L'adaptateur fournira des objets métier pour

quatre des pièces jointes, mais l'ensemble du message est traité comme un événement échoué et archivé dans le dossier des événements échoués, car la pièce jointe trois n'a pas pu être traitée. L'adaptateur consigne les détails relatifs à chaque pièce jointe qu'il ne parvient pas à convertir en objet métier.

Archivage d'événement

Pour archiver les événements, l'adaptateur exige un dossier d'archivage et un dossier réservé aux événements échoués. Les événements sont conservés dans ces dossiers de la façon suivante :

- Si un e-mail est un événement réussi, l'adaptateur le marque FLAGGED, puis le déplace depuis le dossier d'interrogation vers le dossier d'archivage. Si aucun dossier d'archivage n'est spécifié, l'adaptateur supprime l'e-mail du dossier d'interrogation et le message est perdu.
- Si l'e-mail est un événement échoué, l'adaptateur déplace l'e-mail depuis le dossier d'interrogation vers le dossier des événements échoués. Si aucun dossier d'événements échoué n'est précisé, l'adaptateur supprime l'événement du dossier d'interrogation et le message est perdu.

Mécanisme de récupération

Un administrateur peut configurer le mécanisme de récupération de l'adaptateur, pour chaque emplacement d'interrogation précisé par la propriété de configuration PollConfigMO. La propriété PollConfigMO identifie un méta-objet qui dispose d'un attribut InDoubtEvents pour chaque emplacement d'interrogation. Cet attribut peut avoir les valeurs Reprocess, Ignore, LogError et FailOnStartUp.

En cas de panne de l'adaptateur entre les appels d'interrogation, les dossiers d'interrogation peuvent contenir des messages en cours. Ces messages sont marqués d'un indicateur SEEN et considérés comme des événements en attente de validation pour l'adaptateur. Au démarrage suivant de l'adaptateur, la procédure d'initialisation analyse tous les dossiers d'interrogation pour les transactions en attente de validation, de la façon suivante :

- Si la propriété InDoubtEvents de l'emplacement d'interrogation est définie sur Reprocess , l'adaptateur réinitialise l'indicateur SEEN des événements en attente de validation dans le dossier d'interrogation. L'adaptateur prélève ces messages lors des appels d'interrogation suivants.
- Si la propriété InDoubtEvents de l'emplacement d'interrogation est définie sur FailOnStartUp, l'adaptateur arrête l'opération après avoir affiché les messages d'erreur.
- Si InDoubtEvents est défini sur LogError, l'adaptateur continue après avoir affiché les messages d'erreur.
- Si InDoubtEvents est défini sur Ignore, l'adaptateur continue sans traiter les transactions en attente de validation.

Avant de redémarrer l'adaptateur, l'administrateur peut visualiser chacun des dossiers d'interrogation à l'aide d'une programme client de messagerie, et déterminer comment gérer les événements en attente de validation. L'administrateur a le choix entre laisser les messages en attente de validation dans le dossier d'interrogation, réinitialiser leur indicateur SEEN ou supprimer les messages. Lorsque l'adaptateur redémarre, si des événements sont en attente de validation, il traitera les transactions en appliquant la méthode d'initialisation décrite ci-après.

Initialisation de l'adaptateur

Lorsque l'adaptateur pour e-mail démarre, la procédure est la suivante :

1. L'adaptateur extrait la liste de souscription pour les objets métier pris en charge.
2. Grâce aux informations du méta-objet de l'adaptateur, il crée un liste d'emplacements d'interrogation.
3. L'adaptateur tente de se connecter à chaque emplacement d'interrogation. En cas d'échec lors de la connexion à l'un des emplacements d'interrogation, l'adaptateur ne démarre pas.
4. L'adaptateur vérifie que tous les dossiers de chaque emplacement d'interrogation existent. L'adaptateur ne démarre pas si l'un des dossiers n'existe pas.
5. L'adaptateur extrait la valeur de la propriété SMTP_MAILHOST depuis les propriétés de l'adaptateur. A l'aide de cette valeur, il se connecte à l'hôte de messagerie. Si la tentative échoue, l'adaptateur ne démarre pas.
6. L'adaptateur recherche les transactions en attente de validation dans chaque dossier d'interrogation. Si l'adaptateur trouve des transactions en attente de validation, il procède à une reprise sur incident en fonction de la valeur du paramètre InDoubtEvents de l'emplacement d'interrogation. Si l'indicateur InDoubtEvents est défini sur Reprocess, l'adaptateur réinitialise l'indicateur SEEN du message. Si l'indicateur est sur FailOnStartup, l'adaptateur s'arrête. Si l'indicateur est sur LogError, l'adaptateur consigne l'erreur et continue. Si l'indicateur InDoubtEvents est sur Ignore, l'adaptateur continue et ignore les transactions en attente de validation.

Exemple de scénario pour l'adaptateur pour e-mail

La figure 3, à la page 9 illustre un scénario possible. L'illustration montre une solution inter-entreprise qui permet aux partenaires d'échanger des données de gestion par e-mail. Les partenaires ont choisi d'échanger des données de gestion par e-mail plutôt que de communiquer au travers de pare-feu et d'envoyer des données via FTP (File Transfer Protocol).

Un partenaire d'échange A utilise une application professionnelle qui génère des données de gestion. Un WebSphere Business Integration Adapter recherche les événements de l'application, génère des objets métier et les envoie au courtier d'intégration. Ce dernier est configuré pour transmettre les objets métier à l'adaptateur.

L'adaptateur utilise un gestionnaire de données pour convertir les objets métier en un format de données spécifié. L'adaptateur compose l'e-mail, joint les fichiers contenant les données converties et envoie le message par Internet à l'hôte de messagerie MailServer1.

Simultanément, une application propriétaire génère des e-mails contenant les données métier sous forme de chaînes de texte et les envoie à l'hôte de messagerie MailServer2.

Le partenaire d'échange B configure l'adaptateur pour interroger MailServer1 et MailServer2 à la recherche de nouveaux e-mails. Lorsque de nouveaux e-mails arrivent, l'adaptateur contrôle le type MIME de chaque pièce jointe, appelle le gestionnaire de données approprié pour convertir les pièces jointes en objets métier et envoie ceux-ci au courtier d'intégration.

Traitement des données dépendantes des paramètres régionaux

L'adaptateur a été internationalisé, il peut prendre en charge les jeux de caractères à deux octets et transmettre le texte du message dans la langue indiquée. Lorsque l'adaptateur transfère des données entre deux emplacements qui utilisent des jeux différents de codes de caractères, il effectue la conversion des caractères afin de préserver les informations.

L'environnement d'exécution Java de la machine virtuelle Java (JVM) représente les données dans le jeu de codes de caractères Unicode. Le format Unicode contient des codes pour les caractères présents dans la plupart des jeux de codes de caractères connus (à la fois mono-octet et multi-octets). La plupart des composants du système WebSphere Business Integration sont rédigés en Java. Par conséquent, lorsque des données sont transférées entre la plupart des composants du système WebSphere business integration, la conversion des caractères est inutile.

Pour enregistrer les messages d'erreur et d'informations dans la langue et le pays ou territoire approprié, configurez la propriété de configuration standard de l'environnement local pour votre environnement. Pour plus d'informations sur ces propriétés, voir l'Annexe A, «Propriétés standard de configuration pour les adaptateurs», à la page 47.

Caractéristiques du traitement des paramètres régionaux de l'adaptateur pour e-mail

L'adaptateur code les e-mails échangés en fonction du jeu de caractères et des informations de codage configurés au niveau de l'en-tête, du contenu et des objets métier. Les caractéristiques du codage sont les suivantes :

- Dans l'en-tête, seule la ligne d'objet est codée. Les zones indiquant l'émetteur et le destinataire ne le sont pas.
- Les objets métier contenus dans un e-mail peuvent être traités en tant que parties du message ou en tant que pièces jointes. Pour plus d'informations à ce sujet, voir «Structures d'objets métier», à la page 31.

Priorités de configuration des paramètres régionaux

L'adaptateur tente d'appliquer le codage aux données selon un ordre de priorité :

- Pour coder un objet métier contenu dans un e-mail, l'adaptateur contrôle d'abord si le type MIME précisé pour l'objet métier est "text". Dans le cas contraire, l'adaptateur utilise "UTF-8" comme jeu de caractères MIME. Notez que ceci ne s'applique pas à l'en-tête du message ni au contenu, mais seulement à un objet métier.
- L'adaptateur recherche des informations de codage dans l'objet métier d'encapsuleur de premier niveau pour les données qu'il traite. Il s'agit de données sur le contenu et les exigences de routage d'un e-mail. Si les attributs nécessaires ont été définis et configurés, l'adaptateur utilise ces spécifications. Pour plus d'informations sur la structure des objets de premier niveau, voir «Structures d'objets métier», à la page 31.

Si aucune information de codage n'est indiquée au niveau de l'objet métier, l'adaptateur vérifie si les propriétés facultatives sont configurées dans sa propre définition. Si c'est le cas, il réagit en fonction de ces propriétés. Pour plus d'informations sur les propriétés spécifiques à un adaptateur, voir «Propriétés spécifiques à l'adaptateur», à la page 20.

- Si aucune information de codage n'est précisée au niveau de l'objet métier ni de la définition de l'adaptateur, ce dernier utilise les informations de paramètres régionaux par défaut du système.

Common Event Infrastructure

Cet adaptateur est compatible avec IBM Common Event Infrastructure, un standard de gestion des événements qui autorise l'interopérabilité avec d'autres applications IBM WebSphere qui génèrent des événements. Si la prise en charge Common Event Infrastructure est activée, les événements produits par l'adaptateur peuvent être reçus (ou utilisés) par une autre application compatible Common Event Infrastructure.

Pour plus d'informations, voir l'Annexe D, «Common Event Infrastructure», à la page 95.

Application Response Measurement

Cet adaptateur est compatible avec l'interface de programmation Application Response Measurement, qui permet de gérer la disponibilité, les accords de niveau de service et la planification de la capacité des applications. Une application avec appels ARM peut participer à IBM Tivoli Monitoring for Transaction Performance, autorisant ainsi la collecte et la consultation des données relatives aux indicateurs de transaction.

Pour plus d'informations, voir l'Annexe C, «Application Response Measurement», à la page 93.

Chapitre 2. Installation et configuration de l'adaptateur

Ce chapitre décrit comment installer et configurer l'adaptateur de courrier électronique. Il se compose des sections suivantes :

- «Environnement de l'adaptateur pour e-mail»
- «Détermination des tâches d'installation et de configuration», à la page 15
- «Installation de l'adaptateur pour e-mail et des fichiers associés», à la page 17
- «Vérification de l'installation», à la page 17
- «Tâches de configuration de l'adaptateur», à la page 18
- «Configuration de la définition de l'adaptateur», à la page 19
- «Démarrage du connecteur», à la page 24

Environnement de l'adaptateur pour e-mail

Avant d'installer, de configurer et d'utiliser l'adaptateur, vous devez connaître les spécifications inhérentes à son environnement. En complément des sections ci-dessous, les caractéristiques matérielles et logicielles requises sont indiquées dans le Techdoc suivant :

<http://www.ibm.com/support/docview.wss?uid=swg27006249>

Elles font l'objet des sections suivantes :

- «Compatibilité du courtier»
- «Platesformes d'adaptateur», à la page 14
- «Données dépendant des paramètres nationaux», à la page 14

Compatibilité du courtier

L'architecture qu'utilise l'adaptateur doit être compatible avec la version du courtier d'intégration avec lequel l'adaptateur communique. La version 5.4 de l'adaptateur pour e-mail est prise en charge sur les versions de l'architecture d'adaptateur et les courtiers d'intégration suivants :

Architecture de l'adaptateur : WebSphere Business Integration Adapter Framework version 2.7.

Courtiers d'intégration :

- WebSphere InterChange Server, versions 4.2.2, 4.3
- WebSphere MQ Integrator, version 2.1
- WebSphere MQ Integrator Broker, version 2.1
- WebSphere Business Integration Message Broker, version 5.0.1
- WebSphere Application Server Enterprise version 5.0.2, avec WebSphere Studio Application Developer Integration Edition version 5.1.1
- WebSphere Business Integration Server Foundation version 5.1.1

Voir les Notes d'édition pour connaître les exceptions.

Remarque : Pour plus d'informations sur l'installation du courtier d'intégration et les composants requis, voir les documents suivants. Pour WebSphere InterChange Server (ICS), voir le System Installation Guide pour UNIX ou pour Windows.

Pour les courtiers de messages (WebSphere MQ Integrator Broker, WebSphere MQ Integrator et WebSphere Business Integration Message Broker), voir *Implementing Adapters with WebSphere Message Brokers*, et la documentation d'installation du courtier de messages. Certaines de ces informations sont disponibles à l'adresse <http://www.ibm.com/software/integration/mqfamily/library/manualsa/>.

Pour WebSphere Application Server, voir *Implementing Adapters with WebSphere Application Server* et la documentation disponible à l'adresse <http://www.ibm.com/software/webservers/appserv/library.html>.

Platesformes d'adaptateur

Outre le courtier, cet adaptateur exige l'un des systèmes d'exploitation suivants :

- Tous les systèmes d'exploitation exigent le compilateur Java (IBM JDK 1.4.2 for Windows 2000) pour compiler les adaptateurs personnalisés.
- **AIX :**
AIX 5.3
AIX 5.2 avec niveau de maintenance 1. Cet adaptateur prend en charge la JVM 32 bits sur une plateforme 64 bits.
- **Solaris :**
Solaris 8 (2.8) avec le Solaris Patch Cluster du 11 février 2004 ou ultérieur
Solaris 9 (2.9) avec le Solaris Patch Cluster du 11 février 2004 ou ultérieur
Cet adaptateur prend en charge la JVM 32 bits sur une plateforme 64 bits
- **HP-UX :**
HP-UX 11.i (11.11) avec les regroupements GOLDBASE11i et GOLDAPPS11i de juin 2003
- **Linux :**
Red Hat Enterprise Linux AS 3.0 avec Update 1
Red Hat Enterprise Linux ES 3.0 avec Update 1
Red Hat Enterprise Linux WS 3.0 avec Update 1
SUSE Linux Enterprise Server x86 8.1 avec SP3
SUSE Linux Standard Server x86 8.1 avec SP3

Remarque : Le composant TMTP (Tivoli Monitoring for Transaction Performance) de WebSphere Business Integration Adapter FrameworkV2.7 n'est pas pris en charge sous Linux Red Hat.

- **Windows :**
Windows 2000 (Professional, Server ou Advanced Server) avec SP4
Windows XP avec SP1A, pour WebSphere Business Integration Adapter Framework (uniquement pour les outils d'administration)
Windows 2003 (Standard Edition ou Enterprise Edition)

Données dépendant des paramètres nationaux

L'adaptateur a été internationalisé, de sorte qu'il accepte les jeux de caractères à deux octets et peut délivrer le texte du message dans la langue indiquée. Lorsque

l'adaptateur transfère des données entre deux emplacements qui utilisent des jeux différents de codes de caractères, il effectue la conversion des caractères afin de préserver les informations.

L'environnement d'exécution Java dans la machine virtuelle Java (JVM) représente les données dans le jeu de codes de caractères Unicode. Le format Unicode contient des codes pour les caractères présents dans la plupart des jeux de codes de caractères connus (à la fois mono-octet et multi-octets). La plupart des composants du système WebSphere Business Integration sont écrits en Java. Par conséquent, lorsque des données sont transférées entre la plupart des composants du système WebSphere business integration, il n'y a pas lieu de convertir les caractères.

Cet adaptateur prend en charge l'écriture bidirectionnelle pour l'arabe et l'hébreu, lorsqu'il est exécuté dans un environnement Windows. Cette fonctionnalité n'est pas prise en charge dans les autres environnements. Pour enregistrer les messages d'erreur et d'informations dans la langue et le pays ou territoire approprié, configurez la propriété standard de configuration de paramètres régionaux pour votre environnement. Pour plus d'informations sur ces propriétés, voir l'Annexe A, «Propriétés standard de configuration pour les adaptateurs», à la page 47.

Détermination des tâches d'installation et de configuration

Les tâches d'installation et de configuration de l'adaptateur pour e-mail dépendent de ses rôles dans le systèmes WebSphere Business Integration. Les sections ci-dessous vous expliquent comment déterminer les tâches à effectuer, selon la façon dont vous utiliserez l'adaptateur.

Remarque : Si vous installez l'adaptateur pour e-mail en tant que système autonome, consultez les manuels d'installation pour plus d'instructions : *System Installation Guide for Windows* ou *Implementation Guide for WebSphere MQ Integrator Broker*.

Utilisation de l'adaptateur pour la notification d'événement

Si l'adaptateur doit interroger une application pour détecter les nouveaux messages et les envoyer au courtier d'intégration à fin de traitement, procédez comme suit :

1. Installez l'adaptateur comme décrit par «Installation de l'adaptateur pour e-mail et des fichiers associés», à la page 17.
2. Configurez le serveur de messagerie et les boîtes aux lettres pour les emplacements d'interrogation. Pour plus d'informations, voir «Configuration des serveurs de messagerie en vue de l'interrogation», à la page 19.
3. Créez les objets métier nécessaires pour prendre en charge l'interface. Pour plus d'informations, voir le Chapitre 4, «Développement d'objets métier pour l'adaptateur», à la page 31.
4. Créez ou modifiez un méta-objet pour le gestionnaire de données que l'adaptateur utilise pour traiter les données. Pour plus d'informations, voir «Méta-objets du gestionnaire de données», à la page 39.
5. Créez et configurez un méta-objet d'interrogation pour contenir un méta-objet enfant pour chaque emplacement d'interrogation. Pour plus d'informations, voir «Utilisation des méta-objets de l'adaptateur pour e-mail», à la page 40.
6. Configurez les serveurs de messagerie que l'adaptateur va interroger, comme décrit dans «Configuration des serveurs de messagerie en vue de l'interrogation», à la page 19.

7. Configurez la définition de l'adaptateur, comme décrit dans «Configuration de la définition de l'adaptateur», à la page 19. Veillez aux points suivants :
 - Vous devez ajouter les objets métier développés pour l'interface à l'étape 3, en tant qu'objets pris en charge pour l'adaptateur.
 - Vous devez ajouter le méta-objet d'interrogation créé à l'étape 5 en tant qu'objet pris en charge pour l'adaptateur.
 - Vous devez ajouter le méta-objet de gestionnaire de données créé à l'étape 4 en tant qu'objet pris en charge pour l'adaptateur.
 - Vous devez configurer les propriétés de l'adaptateur. Comme il va effectuer des interrogations, vous devez vous assurer de donner à la propriété PollConfigMO le nom du méta-objet d'interrogation créé à l'étape 5.
8. Lancez l'adaptateur comme décrit par «Démarrage du connecteur», à la page 24.

Utilisation de l'adaptateur pour le traitement d'objets métier

Si l'adaptateur doit recevoir des objets métier depuis le courtier d'intégration et les traiter, procédez comme suit :

1. Installez l'adaptateur comme décrit par «Installation de l'adaptateur pour e-mail et des fichiers associés», à la page 17.
2. Créez les objets métier nécessaires pour prendre en charge l'interface. Pour plus d'informations, voir le Chapitre 4, «Développement d'objets métier pour l'adaptateur», à la page 31. Ces objets métier comprendront un objet métier encapsuleur de premier niveau, qui contiendra les informations de routage pour l'adaptateur. Pour plus d'informations, voir «Utilisation des objets métier de l'adaptateur pour e-mail», à la page 31.
3. Créez ou modifiez un méta-objet pour le gestionnaire de données que l'adaptateur utilise pour traiter les données. Pour plus d'informations, voir «Méta-objets du gestionnaire de données», à la page 39.
4. Configurez la définition de l'adaptateur, comme décrit dans «Configuration de la définition de l'adaptateur», à la page 19. Veillez aux points suivants :
 - Vous devez ajouter les objets métier développés pour l'interface à l'étape 3, en tant qu'objets pris en charge pour l'adaptateur.
 - Vous devez ajouter le méta-objet de gestionnaire de données créé à l'étape 4 en tant qu'objet pris en charge pour l'adaptateur.
 - Vous devez configurer les propriétés de l'adaptateur.
5. Lancez l'adaptateur comme décrit par «Démarrage du connecteur», à la page 24.

Utilisation de l'adaptateur pour la notification d'erreur

Remarque : Utilisation de l'adaptateur pour la notification d'erreur (uniquement pour le courtier d'intégration InterChange Server)

Si votre courtier d'intégration est InterChange Server et que l'adaptateur servira pour la notification par courrier électronique, procédez comme suit :

1. Si votre courtier d'intégration est InterChange Server, l'adaptateur pour e-mail est automatiquement installé. En effet, son rôle est tellement important pour la notification d'erreur que l'installation est systématique et vous n'avez pas à vous en occuper. Continuez avec la configuration des propriétés de l'adaptateur, comme décrit par «Configuration de la définition de l'adaptateur», à la page 19.

2. Configurez l'objet métier de notification par courrier électronique comme décrit par «Objet métier EmailNotification», à la page 43.
3. Vous devez configurer les autres composants d'InterChange Server pour la notification par courrier électronique. Pour plus d'informations concernant la configuration de la notification par courrier électronique, voir le manuel *System Administration Guide*.
4. Lancez l'adaptateur comme décrit par «Démarrage du connecteur», à la page 24.

Installation de l'adaptateur pour e-mail et des fichiers associés

Pour plus d'informations sur l'installation des produits de WebSphere Business Integration, voir le guide *Installing WebSphere Business Integration Adapters* dans l'Infocenter de WebSphere Business Integration Adapters, sur le site Web suivant :

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

Vérification de l'installation

Les sections ci-dessous décrivent les chemins et noms des fichiers du produit après l'installation, ainsi que la procédure de vérification de votre installation de l'adaptateur.

Vérification de l'installation sur un système Windows

Avant de commencer : Installez l'adaptateur. L'installation de l'adaptateur copie sur votre système les fichiers standard associés. L'utilitaire installe le connecteur dans le répertoire *ProductDir*\connectors et ajoute un raccourci pour le connecteur dans le menu Démarrer.

Pour vérifier l'installation de l'adaptateur sur un système Windows, procédez comme suit : Passez dans le répertoire d'installation *ProductDir*\ et comparez son contenu avec celui indiqué dans le Tableau 2.

Le Tableau 2 décrit la structure de fichiers utilisée par l'adaptateur sous Windows, et indique les fichiers qui sont automatiquement installés avec lui lorsque vous utilisez le programme d'installation.

Tableau 2. Structure des fichiers installés sous Windows avec le connecteur

Sous-répertoire de <i>ProductDir</i>	Description
\connectors\EMail	Contient les fichiers de classe de composant spécifiques à l'adaptateur CWEmail.jar, ainsi que start_Email.bat, le fichier de démarrage de l'adaptateur.
\connectors\messages	Contient les codes d'erreur et la descriptions des erreurs utilisées par le connecteur.
\connectors\EMail\samples	Contient les fichiers utilisés pour tester le connecteur.
\connectors\EMail\JavaMail-Sun	Contient le fichier mail.jar de JavaMail et le fichier activation.jar de Java Activation Framework.

Remarque : Tous les noms de chemin des produits sont associés au répertoire dans lequel le produit est installé sur votre système.

Vérification de l'installation sur un système Unix

Avant de commencer : Installez l'adaptateur. L'installation de l'adaptateur copie sur votre système les fichiers standard associés. L'utilitaire installe l'agent connecteur dans le répertoire *ProductDir/connectors*.

Pour vérifier l'installation de l'adaptateur sur un système UNIX, procédez comme suit : Passez dans le répertoire d'installation *ProductDir/* et comparez son contenu avec celui indiqué dans le Tableau 3.

Le Tableau 3 décrit la structure de fichiers utilisée par l'adaptateur sous UNIX, et indique les fichiers qui sont automatiquement installés avec lui lorsque vous utilisez le programme d'installation.

Tableau 3. Structure de fichiers installée sous UNIX pour le connecteur

Sous-répertoire de <i>ProductDir</i>	Description
<i>/connectors/EMail</i>	Contient les fichiers de classe de composant spécifiques à l'adaptateur <i>CWEmail.jar</i> , ainsi que <i>start_Email.sh</i> , le fichier de démarrage de l'adaptateur.
<i>/connectors/messages</i>	Contient les codes d'erreur et la descriptions des erreurs utilisées par le connecteur.
<i>/connectors/EMail/samples</i>	Contient les fichiers utilisés pour tester le connecteur.
<i>/connectors/EMail/JavaMail-Sun</i>	Contient le fichier <i>mail.jar</i> de JavaMail et le fichier <i>activation.jar</i> de Java Activation Framework.

Remarque : Tous les noms de chemin des produits sont associés au répertoire dans lequel le produit est installé sur votre système.

Tâches de configuration de l'adaptateur

Les tâches de configuration de l'adaptateur pour e-mail sont nombreuses, mais toutes ne sont pas nécessaires, selon l'usage prévu pour l'adaptateur. Voir «Détermination des tâches d'installation et de configuration», à la page 15 pour déterminer parmi les tâches suivantes celles qui sont nécessaires dans votre cas, puis les sections ci-dessous pour obtenir des informations sur une tâche donnée.

Cette section aborde les points suivants :

- «Configuration des serveurs de messagerie en vue de l'interrogation», à la page 19
- «Configuration de la définition de l'adaptateur», à la page 19
- «Propriétés standard de configuration», à la page 19
- «Propriétés spécifiques à l'adaptateur», à la page 20

Configuration des serveurs de messagerie en vue de l'interrogation

Si l'adaptateur pour e-mail doit interroger pour obtenir les messages à envoyer au courtier d'intégration, vous devez configurer les serveurs de messagerie pour indiquer les boîtes aux lettres et les emplacements d'interrogation que l'adaptateur utilisera. Procédez comme suit :

1. Créez les comptes utilisateur qui peuvent accéder aux emplacements d'interrogation. Les utilisateurs peuvent être sur un seul ou sur plusieurs serveurs de messagerie.
2. Configurez les emplacements d'interrogation pour chaque utilisateur, en créant pour chaque emplacement un dossier d'interrogation, un dossier d'archive et un dossier d'événements d'échec. Bien qu'un utilisateur puisse avoir plusieurs emplacements d'interrogation, et que chaque emplacement puisse être sur un serveur de messagerie différent, les trois dossiers d'un emplacement d'interrogation doivent être dans la même boîte aux lettres. L'adaptateur se connecte à ces dossiers avec le nom d'utilisateur et le mot de passe du propriétaire de la boîte aux lettres.

L'adaptateur n'utilise pas cette boîte aux lettres lorsqu'il envoie des demandes d'objets métier. Pour cela, il utilise le nom du serveur de messagerie indiqué dans sa propriété SMTP_MailHost.

Configuration de la définition de l'adaptateur

Avant de lancer l'adaptateur pour e-mail, vous devez le configurer. Procédez comme suit :

- Définissez les propriétés standard et celles qui sont spécifiques à l'adaptateur.
- Si l'adaptateur doit interroger pour obtenir les événements, vous devez configurer les méta-objets requis. Voir «Utilisation des méta-objets de l'adaptateur pour e-mail», à la page 40 pour la configuration des méta-objets.

Une fois ces méta-objets configurés, vous devez ajouter leur prise en charge à la définition de l'adaptateur. Pour configurer une définition d'adaptateur, vous utiliserez Connector Configurator (que vous lancerez depuis System Manager si votre courtier d'intégration est ICS), ou Connector Configurator (si votre courtier d'intégration est WebSphere MQ Integrator Broker).

L'adaptateur obtient ses valeurs de configuration lors du démarrage. En cours d'exécution, vous pourrez avoir besoin de changer la valeur d'une ou de plusieurs propriétés. Selon la propriété et le système d'intégration, ces changements peuvent prendre effet immédiatement, nécessiter un redémarrage de l'adaptateur ou bien un redémarrage complet du système d'intégration. Pour savoir dans quelles conditions les modifications prennent effet, consultez la colonne Update Method de l'onglet Properties approprié, dans l'outil de configuration de l'adaptateur.

Propriétés standard de configuration

Les propriétés standard de configuration fournissent des informations utilisées par tous les connecteurs. Voir l'Annexe A, «Propriétés standard de configuration pour les adaptateurs», à la page 47 pour plus d'informations sur ces propriétés.

Important : Comme cet adaptateur prend en charge tous les courtiers d'intégration, les propriétés de configuration de tous les courtiers lui sont applicables.

Le tableau 4 donne des informations spécifiques à cet adaptateur et concernant les propriétés de configuration de l'annexe.

Tableau 4. Informations sur les propriétés spécifiques à cet adaptateur

Propriété	Important
CharacterEncoding ApplicationName	Cet adaptateur n'utilise pas cette propriété. Avant de lancer l'adaptateur, vous devez indiquer une valeur pour cette propriété de configuration.
Locale	Cet adaptateur étant internationalisé, vous pouvez modifier la valeur de cette propriété. Voir les notes d'informations concernant l'adaptateur pour connaître les paramètres régionaux pris en charge.

Propriétés spécifiques à l'adaptateur

Les propriétés de configuration spécifiques à l'adaptateur donnent des informations spécifiques à l'application ou à la technologie intégrée par l'adaptateur. Vous pouvez utiliser ces propriétés pour modifier le comportement de l'adaptateur sans avoir à recoder et reconstituer le composant spécifique à l'application.

Le tableau 5 présente les propriétés de configuration spécifiques à l'adaptateur pour e-mail. Pour obtenir une explication des propriétés, voir les sections suivantes.

Tableau 5. propriétés de configuration spécifiques à l'adaptateur

Nom	Valeurs possibles	Valeur par défaut	Obligatoire
ApplicationPassword	le mot de passe correspondant à ApplicationUserName		Oui, si ApplicationUserName est défini
ApplicationUserName	le nom d'utilisateur requis par le serveur SMTP		Non
BusinessObjectMimeCharset	une valeur valide pour un jeu de caractères		Non
BusinessObjectMimeEncoding	une valeur valide pour un codage de données		Non
DataHandlerConfigMO	le nom du méta-objet du gestionnaire de données	MO_DataHandler_Default	Oui
DebugMode	true ou false	false	Non
DefaultVerb	une instruction valide pour les objets métier gérés par l'adaptateur	Create	Non
InboundWrapperBO	le nom de l'objet métier encapsuleur pour les communications entrantes		Non
MailsPerMailBox	Un entier positif	1	Non
MessageContentMimeCharset	une valeur valide pour un jeu de caractères		Non
MessageContentMimeEncoding	une valeur valide pour un codage de données		Non
MessageHeaderMimeCharset	une valeur valide pour un jeu de caractères		Non
MessageHeaderMimeEncoding	une valeur valide pour un codage de données		Non
PollConfigMO	le nom du méta-objet		Non
PollQuantity	Un entier positif	1	Non
SMTP_MailPort	Le numéro de port pour l'hôte SMTP		Non
UseDefaults	définit la valeur par défaut pour un attribut obligatoire	False	Non

ApplicationPassword

le mot de passe qui est associé à la valeur indiquée pour ApplicationUserName.

ApplicationUserName

Le nom d'utilisateur requis pour l'authentification par le serveur SMTP.

BusinessObjectMimeCharset

Le jeu de caractères des objets métier contenus dans l'e-mail. Voir les RFC d'e-mail pour déterminer la valeur convenant à l'environnement. Il peut s'agir par exemple de iso-8859-1, le jeu de caractères Latin 1 pour les langues d'Europe de l'Ouest.

BusinessObjectMimeEncoding

Le codage des objets métier contenus dans l'e-mail. Voir les RFC d'e-mail pour déterminer la valeur convenant à l'environnement. Par exemple, vous indiqueriez Q pour signifier "Quoted-Printable" ou B pour "Base64". Si cette propriété est définie, les objets métier sont envoyés comme parties du message. Si elle n'est pas définie, ils sont envoyés comme fichiers joints.

DataHandlerConfigMO

Le nom du méta-objet utilisé par le gestionnaire de données pour définir les propriétés de configuration. Sert également à la classe de base `DataHandler` pour déterminer quelle classe `DataHandler` utiliser pour un type particulier de contenu. Voir «[Meta-objets du gestionnaire de données](#)», à la page 39 pour plus d'informations sur ce méta-objet.

DebugMode

Indique si les messages de débogage JavaMail sont envoyés sur `STDOUT`.

DefaultVerb

Indique la valeur insérée dans l'attribut `Verb` pour les objets métier de premier niveau créés par l'adaptateur pendant la notification d'événement, si cet attribut est vide. Dans certains cas, l'adaptateur peut interroger un événement et créer un objet métier, mais la source de données (comme un document XML) peut ne pas contenir d'informations qui corresponde directement à l'attribut `Verb` de la définition d'objet métier. Si l'objet métier est envoyé au courtier d'intégration sans une instruction (`Verb`) valide, il est considéré comme sans abonnement, et n'est pas traité. Vous pouvez indiquer ici une instruction valide (comme `Create`), afin de remplir cette propriété au cas où l'attribut `Verb` serait vide ou de valeur `null`.

La valeur par défaut est `Create`.

InboundWrapperBO

Indique au connecteur d'assembler tous les objets métier créés à partir des pièces jointes, et de les envoyer ensemble au courtier d'intégration. La structure de l'objet métier possède également des attributs comme `FromAddress` et `ToAddress`, qui sont renseignés par le connecteur.

MailsPerMailBox

Indique le nombre de messages traités dans chaque boîte aux lettres avant que l'adaptateur passe au traitement de la boîte aux lettres suivante.

MessageContentMimeCharset

Le jeu de caractères du contenu du message. Voir les RFC d'e-mail pour déterminer la valeur convenant à l'environnement. Il peut s'agir par exemple de `iso-8859-1`, le jeu de caractères Latin 1 pour les langues d'Europe de l'Ouest.

MessageContentMimeEncoding

Le codage du contenu du message. Voir les RFC d'e-mail pour déterminer la valeur convenant à l'environnement. Par exemple, vous indiqueriez `Q` pour signifier "Quoted-Printable" et `B` pour "Base64".

MessageHeaderMimeCharset

Le jeu de caractère de l'en-tête du message. Voir les RFC d'e-mail pour déterminer la valeur convenant à l'environnement. Il peut s'agir par exemple de `iso-8859-1`, le jeu de caractères Latin 1 pour les langues d'Europe de l'Ouest.

MessageHeaderMimeEncoding

Le codage de l'en-tête du message. Voir les RFC d'e-mail pour déterminer la valeur convenant à l'environnement. Par exemple, vous indiqueriez `Q` pour signifier "Quoted-Printable" ou `B` pour "Base64".

PollConfigMO

Le nom du méta-objet utilisé par l'adaptateur pour e-mail pour ses interrogations. Pour plus d'informations sur le méta-objet adaptateur pour e-mail, voir «[Utilisation des méta-objets de l'adaptateur pour e-mail](#)», à la page 40.

PollQuantity

Indique le nombre maximal d'événements interrogés pendant un seul appel d'interrogation.

SMTP_MailPort

Le numéro de port utilisé par l'hôte de courrier SMTP pour envoyer le courrier électronique.

UseDefaults

Lorsque cette propriété est définie sur true, les attributs obligatoires recevront une valeur par défaut. Lorsque cette propriété est définie sur false, les valeurs par défaut ne seront pas utilisées. La valeur par défaut est False.

Remarque : Si vous avez l'intention d'utiliser la valeur par défaut pour l'attribut FromAddress au niveau de la définition de l'objet métier EmailNotification (selon votre configuration, il se peut que votre objet métier Email Notification n'ait pas de valeur pour FromAddress), vous devez définir UseDefaults sur true. Sinon, l'adaptateur signalera des erreurs.

Création de plusieurs instances de connecteur

La création de plusieurs instances d'un connecteur revient pratiquement à créer un connecteur personnalisé. Vous pouvez configurer votre système de sorte qu'il crée et exécute plusieurs instances d'un connecteur en suivant les étapes ci-dessous.

Vous devez :

- créer un répertoire pour l'instance du connecteur ;
- vérifier que vous possédez les définitions d'objet métier requises ;
- créer un fichier de définition pour le connecteur ;
- créer un script de démarrage.

Création d'un répertoire

Vous devez créer un répertoire pour chaque instance de connecteur. Vous devez attribuer le nom suivant à ce répertoire de connecteur :

`ProductDir\connectors\connectorInstance`

où connectorInstance identifie de manière unique l'instance de connecteur.

Si le connecteur possède des méta-objets qui lui sont spécifiques, vous devez créer un méta-objet pour l'instance de connecteur. Si vous enregistrez le méta-objet en tant que fichier, créez le répertoire suivant et stockez le fichier dedans :

`ProductDir\Repository\connectorInstance`

Création de définitions d'objet métier

Si les définitions d'objet métier pour chaque instance du connecteur n'existent pas déjà dans le projet, vous devez les créer.

1. Si vous devez modifier les définitions d'objet métier associées au connecteur initial, copiez les fichiers appropriés et utilisez Business Object Designer pour les importer. Vous pouvez copier n'importe quel fichier pour le connecteur initial. Vous devez simplement les renommer si vous les modifiez.
2. Les fichiers pour le connecteur initial doivent résider dans le répertoire suivant :

`ProductDir\Repository\initialConnectorInstance`

Tous les fichiers supplémentaires que vous créez doivent être placés dans le sous-répertoire `connectorInstance` approprié de `ProductDir\repository`.

Création d'une définition de connecteur

Vous devez créer un fichier de configuration (définition du connecteur) pour l'instance du connecteur dans Connector Configurator. Pour ce faire, procédez comme suit :

1. Copiez le fichier de configuration du connecteur initial (définition du connecteur) et renommez-le.
2. Assurez-vous que chaque instance du connecteur répertorie correctement ses objets métier pris en charge (et tous les méta-objets associés).
3. Personnalisez toutes les propriétés du connecteur le cas échéant.

Création d'un script de démarrage

Pour créer un script de démarrage, procédez comme suit :

1. Copiez le script de démarrage du connecteur initial et attribuez-lui un nom incluant le nom du répertoire du connecteur :
`dirname`
2. Placez ce script de démarrage dans le répertoire du connecteur créé à la section «Création d'un répertoire», à la page 23.
3. Créez un raccourci pour le script de démarrage (Windows uniquement).
4. Copiez le texte du raccourci du connecteur et modifiez le nom du connecteur initial (dans la ligne de commande) de sorte qu'il corresponde au nom de la nouvelle instance du connecteur.

A présent, vous pouvez exécuter simultanément les deux instances du connecteur sur votre serveur d'intégration.

Pour plus d'informations sur la création de connecteurs personnalisés, voir *Connector Development Guide for C++ or for Java*.

Démarrage du connecteur

Vous devez démarrer un connecteur de manière explicite à l'aide du **script de démarrage du connecteur**. Sous Windows, le script de démarrage doit résider dans le répertoire d'exécution du connecteur :

`ProductDir\connectors\connName`

où `connName` identifie le connecteur.

Sous UNIX, le script de démarrage doit résider dans le répertoire *UNIX* `ProductDir/bin`.

Le nom du script de démarrage dépend de la plateforme du système d'exploitation, comme le montre le tableau 6.

Tableau 6. Script de démarrage pour un connecteur

Système d'exploitation	Script de démarrage
Systèmes UNIX	<code>connector_manager</code>
Windows	<code>start_connName.bat</code>

Lorsque le script de démarrage s'exécute, il va chercher par défaut le fichier de configuration dans le *ProductDir* (voir commandes ci-dessous). Il s'agit du répertoire dans lequel vous placez le fichier de configuration.

Remarque : Si l'adaptateur utilise le transfert JMS, vous avez besoin d'un fichier de configuration local .

Pour appeler le script de démarrage du connecteur, utilisez l'une des méthodes suivantes :

- Sur les systèmes Windows, dans le menu **Démarrer** :
Sélectionnez **Programmes>IBM WebSphere Business Integration Adapters>Adapters>Connectors**. Par défaut, le nom du programme est "IBM WebSphere Business Integration Adapters". Cependant, vous pouvez le personnaliser. Vous pouvez également créer sur le bureau un raccourci vers le connecteur.

- A partir de la ligne de commande :

– Sur les systèmes Windows :

```
start_connName connName brokerName [-cconfigFile]
```

– Sur les systèmes UNIX :

```
connector_manager -start connName  
brokerName [-cconfigFile ]
```

où *connName* est le nom du connecteur et *brokerName* le nom de votre connecteur d'intégration, comme suit :

- Pour WebSphere InterChange Server, indiquez à la place de *brokerName* le nom de l'instance ICS.
- Pour les courtiers de messages WebSphere (WebSphere MQ Integrator, WebSphere MQ Integrator Broker ou WebSphere Business Integration Message Broker) ou WebSphere Application Server, remplacez *brokerName* par une chaîne identifiant le courtier.

Remarque : Pour un courtier de messages WebSphere ou WebSphere Application Server résidant sur un système Windows, vous devez inclure l'option *-c* suivie du nom du fichier de configuration du connecteur. Pour ICS, l'option *-c* est facultative.

- A partir de Adapter Monitor, qui est lancé au démarrage de System Manager, lequel est exécuté avec le courtier WebSphere Application Server ou InterChange Server :

Vous pouvez charger, activer, désactiver, interrompre, arrêter ou supprimer un connecteur à l'aide de cet outil.

- A partir de System Manager (pour tous les courtiers) :

Vous pouvez charger, activer, désactiver, interrompre, arrêter ou supprimer un connecteur à l'aide de cet outil.

- Sur les systèmes Windows, vous pouvez configurer le connecteur de sorte qu'il démarre comme un service Windows. Dans ce cas, le connecteur démarre lors de l'amorçage du système Windows (pour un service automatique) ou lorsque vous démarrez le service via la fenêtre Services Windows (pour un service manuel).

Pour plus d'informations sur le démarrage d'un connecteur, notamment sur les options de lancement à partir de la ligne de commande, reportez-vous à l'un des documents suivants :

- Pour WebSphere InterChange Server, voir *System Administration Guide*.

- Pour les courtiers de messages WebSphere, voir *Implementing Adapters with WebSphere Message Brokers*.
- Pour WebSphere Application Server, voir *Implementing Adapters with WebSphere Application Server*.

Arrêt du connecteur

La méthode pour arrêter un connecteur dépend de la manière dont il a été démarré, comme suit :

- Si vous avez démarré le connecteur à partir d'une ligne de commande, avec son script de démarrage :
 - Sur les systèmes Windows, l'appel du script de démarrage crée une fenêtre de "console" séparée pour le connecteur. Dans cette fenêtre, tapez "Q" et appuyez sur Entrée pour arrêter le connecteur.
 - Avec InterChange Server sur les systèmes UNIX, les connecteurs s'exécutent en arrière-plan de sorte qu'ils n'ont pas de fenêtre séparée. Vous devez exécuter la commande suivante pour arrêter le connecteur :

```
connector_manager_connName -stop
```

où *connName* correspond au nom du connecteur.
- A partir d'Adapter Monitor (produit WebSphere Business Integration Adapters uniquement), lancé lorsque vous démarrez System Manager :
Vous pouvez charger, activer, désactiver, interrompre, arrêter ou supprimer un connecteur à l'aide de cet outil.
- A partir de System Monitor (produit WebSphere InterChange Server uniquement) :
Vous pouvez charger, activer, désactiver, interrompre, arrêter ou supprimer un connecteur à l'aide de cet outil.
- Sur les systèmes Windows, vous pouvez configurer le connecteur de sorte qu'il démarre comme un service Windows. Dans ce cas, le connecteur s'arrête en même temps que le système Windows.

Chapitre 3. Planification de l'implémentation d'un adaptateur pour e-mail

L'adaptateur pour e-mail assure l'intégration des applications dans les limites des pare-feu et au-delà, grâce aux messages électroniques. Ce chapitre fournit des informations sur la planification d'une implémentation de l'adaptateur pour e-mail. Il contient les sections suivantes :

- «Présentation de l'implémentation d'un adaptateur pour e-mail»
- «Fonctionnement de l'adaptateur pour e-mail et des gestionnaires de données», à la page 28
- «Exemples de fichiers de l'adaptateur pour e-mail», à la page 29

Présentation de l'implémentation d'un adaptateur pour e-mail

Le processus d'implémentation de l'adaptateur pour e-mail implique diverses tâches. Voici une présentation des tâches à effectuer, avant d'utiliser l'adaptateur pour e-mail pour envoyer et recevoir des objets métier :

1. Déterminez le format des données qui seront envoyées et reçues dans les pièces jointes des e-mails.
2. Une fois les formats des données identifiés, vous pouvez déterminer les gestionnaires de données qui seront utilisés par l'adaptateur pour convertir les objets métier en e-mails et inversement. Pour plus d'informations sur les gestionnaires de données, voir «Fonctionnement de l'adaptateur pour e-mail et des gestionnaires de données», à la page 28.
3. Si l'adaptateur doit gérer des requêtes d'objet métier, créez les définitions d'objet métier qui seront utilisées par l'adaptateur pour e-mail. Pour plus d'informations, voir le Chapitre 4, «Développement d'objets métier pour l'adaptateur», à la page 31.
4. Pour envoyer et recevoir les objets métier souhaités par e-mail, configurez l'adaptateur pour e-mail avec les autres composants du courtier d'intégration.
5. Si votre courtier d'intégration est InterChange Server, vous devrez peut-être modifier le méta-objet de premier niveau du gestionnaire de données. Pour les adaptateurs, le méta-objet de premier niveau par défaut se nomme MO_DataHandler_Default. Pour modifier ce méta-objet, utilisez Business Object Designer. Pour plus d'informations sur les méta-objets de gestionnaire de données, voir «Méta-objets du gestionnaire de données», à la page 39.
6. Si vous devez implémenter une interrogation, déterminez à quels serveurs de messagerie l'adaptateur se connectera, et créez les boîtes aux lettres des emplacements d'interrogation. Pour plus d'informations, voir «Configuration des serveurs de messagerie en vue de l'interrogation», à la page 19.
7. Pour l'interrogation, configurez le méta-objet de l'adaptateur pour e-mail. Il définit les emplacements d'interrogation de l'adaptateur pour messagerie. Pour plus d'informations, voir «Utilisation des méta-objets de l'adaptateur pour e-mail», à la page 40.
8. Configurez l'adaptateur en définissant ses propriétés. Par exemple, indiquez le nom du méta-objet du gestionnaire de données de premier niveau dans l'attribut DataHandlerConfigMO de configuration de l'adaptateur pour e-mail. Pour plus d'informations sur les propriétés de configuration de l'adaptateur, voir «Configuration de la définition de l'adaptateur», à la page 19.

Fonctionnement de l'adaptateur pour e-mail et des gestionnaires de données

L'adaptateur pour e-mail est conçu pour utiliser les gestionnaires de données afin de convertir les objets métier vers les formats précisés, et de transformer en objets métier les données aux formats précisés. Le gestionnaire de données n'est pas intégré à l'adaptateur, c'est un module séparé qui peut être personnalisé ou remplacé au besoin.

Vous pouvez utiliser les gestionnaires de données fournis par IBM ou écrire les vôtres. Si les données sont dans un format propriétaire ou si les gestionnaires de données fournis par IBM ne prennent pas en charge le format de données de l'application, vous devrez écrire un gestionnaire de données personnalisé. Par exemple, IBM fournit un gestionnaire de données qui procède à la conversion entre les objets métier et les données XML. Si votre application génère des documents HTML plutôt que des documents XML, vous devrez implémenter un gestionnaire de données personnalisé qui procédera à la conversion entre les données HTML et les objets métier.

Avant de commencer à utiliser l'adaptateur pour e-mail, prenez le temps d'analyser les formats de données qui seront pris en charge par votre implémentation. Utilisez ensuite les informations du tableau 7 pour savoir si vous pouvez utiliser les gestionnaires de données fournis par IBM.

Tableau 7. Gestionnaires de données fournis par WebSphere Business Integration

Gestionnaires de données fournis par IBM	Description
Gestionnaire de données XML	Convertit les objets métier depuis et vers les documents XML. Prend en charge XML 1.0.
Gestionnaire de données NameValue	Procède à l'analyse syntaxique des données texte en fonction des zones nommées, telles que celles qui identifient le type d'objet métier.
Gestionnaire de données FixedWidth	Procède à l'analyse syntaxique du texte en utilisant des zones de longueur fixe. Les longueurs des zones sont précisées par la propriété MaxLength de chaque attribut d'objet métier.
Gestionnaire de données délimité	Procède à l'analyse syntaxique des données texte en fonction d'un délimiteur spécifié qui sépare les zones individuelles des données de l'objet métier.

Pour plus d'informations sur les gestionnaires de données, voir le *Data Handler Guide*. Si vous devez écrire un gestionnaire de données personnalisées, voir le *Data Handler Guide* pour savoir comment procéder.

Exemples de fichiers de l'adaptateur pour e-mail

Vous souhaitez peut-être charger et exécuter les fichiers d'exemple de l'adaptateur pour e-mail afin de consulter un exemple d'utilisation de l'adaptateur. L'exemple présente un scénario dans lequel l'e-mail est utilisé pour intégrer des applications à travers un pare-feu.

Les fichiers incluent des objets métier, des collaborations (à utiliser avec le courtier d'intégration InterChange Server), et un fichier readme contenant des instructions de configuration et d'exécution de l'exemple. Les fichiers d'exemple sont situés dans `connector\Email\samples`. Pour exécuter l'exemple, vous devrez configurer un serveur de messagerie et un programme de messagerie client.

Chapitre 4. Développement d'objets métier pour l'adaptateur

Ce chapitre décrit la structure des objets métier requis par l'adaptateur pour e-mail. Il contient les sections suivantes :

- «Utilisation des objets métier de l'adaptateur pour e-mail»
- «Meta-objets du gestionnaire de données», à la page 39
- «Utilisation des méta-objets de l'adaptateur pour e-mail», à la page 40
- «Objet métier EmailNotification», à la page 43
- «Indication d'adresses e-mail valides», à la page 45

Utilisation des objets métier de l'adaptateur pour e-mail

En utilisant un adaptateur pour e-mail, un courtier d'intégration peut envoyer un message par courrier électronique et y attacher des objets métier ou des fichiers. Les objets métier ou fichiers sont encapsulés dans un objet métier hiérarchique de premier niveau, qui contient les informations de routage de l'adaptateur. Les objets métier à envoyer sont définis en tant qu'enfants de l'objet métier de premier niveau. Si l'adaptateur pour e-mail est utilisé pour traiter les requêtes d'objet métier, vous devez préciser ces définitions d'objet métier d'encapsuleur.

Notez que les objets métier d'encapsuleur n'ont besoin d'être définis que pour le traitement des requêtes. Ils ne sont pas utilisés pour l'interrogation. Lors d'une interrogation, l'adaptateur examine le type MIME de chacune des pièces jointes au courrier électronique, instancie le gestionnaire de données approprié, et transmet la pièce jointe au gestionnaire de données. Ce dernier examine le nom de l'objet métier depuis la pièce jointe, trouve la définition de l'objet métier correspondant dans la liste des objets métier pris en charge pour l'adaptateur, et convertit le contenu de la pièce jointe en un objet métier.

Structures d'objets métier

L'objet métier de premier niveau de l'adaptateur pour e-mail doit contenir un ensemble d'attributs décrivant le contenu et les informations de routage d'un message électronique. L'adaptateur compose le message électronique à l'aide des attributs obligatoires d'objet métier, indiqués dans le tableau 8.

Tableau 8. Attributs obligatoires dans un objet métier de premier niveau pour e-mail

Nom	Description
RecipientName	Adresse(s) électronique(s) du destinataire. Pour plus d'informations, voir «Indication d'adresses e-mail valides», à la page 45.
MessageSubject	Description du message électronique. Cette valeur est définie par le courtier d'intégration dans l'objet métier.
MessageContent	Contenu du message électronique. Cette valeur est définie par le courtier d'intégration dans l'objet métier.

Tableau 8. Attributs obligatoires dans un objet métier de premier niveau pour e-mail (suite)

Nom	Description
FromAddress	Adresse électronique de l'émetteur. L'émetteur reçoit un avis de non-livraison de message. Pour plus d'informations, voir «Indication d'adresses e-mail valides», à la page 45. Si vous utilisez la valeur par défaut de cet attribut, vous devez également définir la propriété du connecteur UseDefaults sur true. Pour plus d'informations sur la propriété UseDefaults, voir le tableau 5, à la page 21.
MimeType	Type mime des pièces jointes. S'il n'est pas indiqué, l'adaptateur ne traite pas les objets métier enfants.
AttachmentExtension	Définissez la propriété Default Value de cet attribut sur l'extension qui sera celle des fichiers représentant les objets métier contenus dans le courrier électronique. Par exemple, si un objet métier est envoyé en pièce jointe à un message électronique, et si la propriété Default Value de cet attribut est définie sur txt , la pièce jointe qui représente l'objet métier aura l'extension .txt.
BusinessObjectMimeType	Le jeu de caractères des objets métier contenus dans le courrier électronique. Pour plus d'informations sur cette propriété, voir sa description au niveau de la définition de l'adaptateur, dans la section «BusinessObjectMimeType», à la page 21.
BusinessObjectMimeTypeEncoding	Le codage des objets métier contenus dans le courrier électronique. Pour plus d'informations sur cette propriété, voir sa description au niveau de la définition de l'adaptateur, dans la section «BusinessObjectMimeTypeEncoding», à la page 21.
ContentType	Type de contenu du message à envoyer. Les valeurs valides sont text/plain.
DataEncoding	Type de message : text ou binary.
MessageContentMimeType	Le jeu de caractères du contenu du message. Pour plus d'informations sur cette propriété, voir sa description au niveau de la définition de l'adaptateur, dans la section «MessageContentMimeType», à la page 22.
MessageContentMimeTypeEncoding	Le codage du contenu du message. Pour plus d'informations sur cette propriété, voir sa description au niveau de la définition de l'adaptateur, dans la section «MessageContentMimeTypeEncoding», à la page 22.
MessageHeaderMimeType	Le jeu de caractère de l'en-tête du message. Pour plus d'informations sur cette propriété, voir sa description au niveau de la définition de l'adaptateur, dans la section «MessageHeaderMimeType», à la page 22.

Tableau 8. Attributs obligatoires dans un objet métier de premier niveau pour e-mail (suite)

Nom	Description
MessageHeaderMimeEncoding	Le codage de l'en-tête du message. Pour plus d'informations sur cette propriété, voir sa description au niveau de la définition de l'adaptateur, dans la section «MessageHeaderMimeEncoding», à la page 22.
ReplyTo	Adresse(s) ReplyTo de l'en-tête du message électronique. Pour plus d'informations, voir «Indication d'adresses e-mail valides», à la page 45.

En plus des attributs indiqués dans le tableau 8, l'objet métier de premier niveau peut contenir les attributs facultatifs répertoriés dans le tableau 9.

Tableau 9. Attributs facultatifs d'un objet métier de premier niveau pour e-mail

Nom	Description
PriorityLevel	Les valeurs valides sont Normal, High et Low. L'adaptateur utilise cette valeur pour déterminer la priorité du message sortant.
ContentType	Les valeurs valides sont text/plain et text/html. La valeur par défaut est text/plain.
DataEncoding	Type de message. Soit text soit binary. La valeur par défaut est text.
attributs qui représentent les objets métier hiérarchiques	L'adaptateur convertit les objets métier enfants en pièces jointes à un message électronique.
attributs qui spécifient les noms des fichiers	L'adaptateur inclut les fichiers en tant que pièces jointes au message électronique.

La figure 4 illustre la structure de base d'un objet métier de premier niveau, nommé Email_TLO_Customer_Wrapper, pour l'adaptateur pour e-mail. Cet objet métier contient un objet métier Customer hiérarchique en tant qu'enfant, et inclut un attribut pointant vers un fichier. L'objet métier Customer et le fichier sont ajoutés au courrier électronique en tant que pièces jointes.

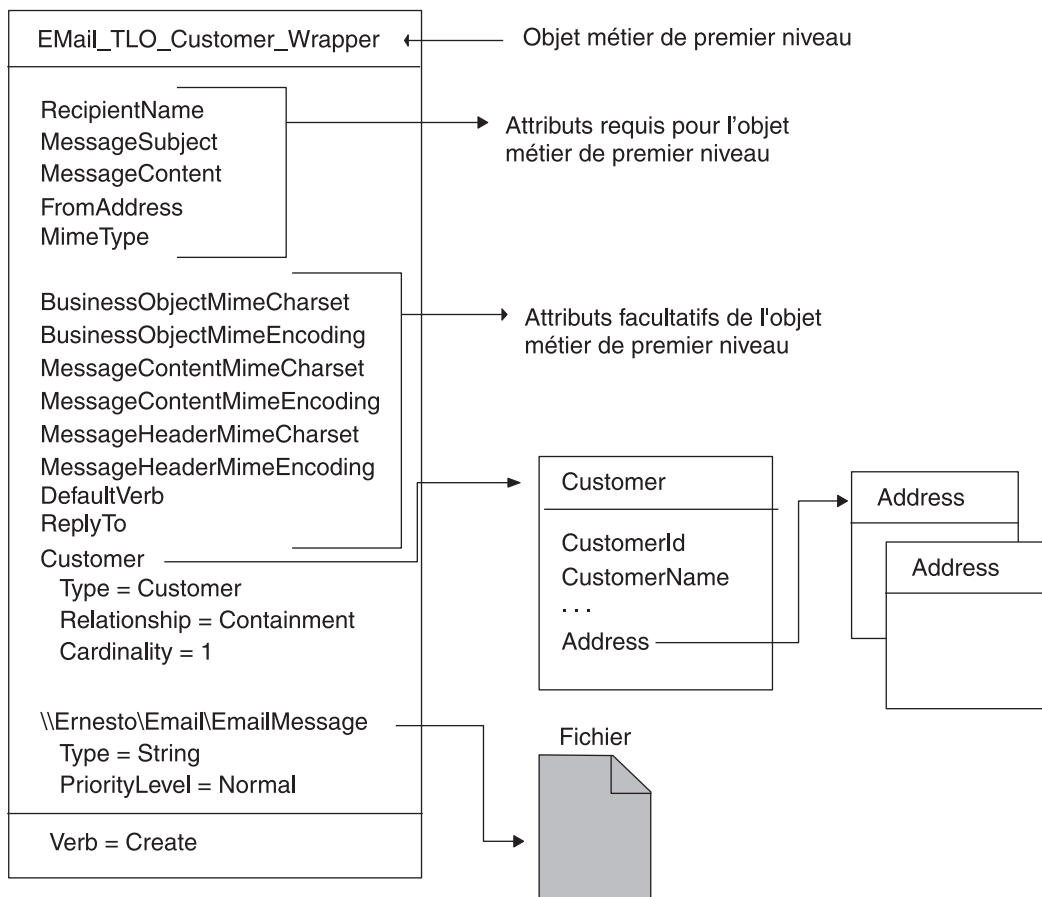


Figure 4. Exemple d'objet métier d'adaptateur pour e-mail

Structure facultative d'objet métier enfant pour la personnalisation des noms des pièces jointes et l'envoi de pièces jointes binaires

Par défaut, l'adaptateur pour e-mail crée une pièce jointe nommée d'après l'objet métier envoyé pour demander le traitement, et complète le message électronique avec les informations obtenues dans les listes d'attributs obligatoires et facultatifs indiquées ci-dessus. Toutefois, pour créer un nom de pièce jointe personnalisé ou inclure une pièce jointe binaire, chaque objet métier doit utiliser un objet métier enfant pour préciser les informations relatives à la pièce jointe. Voir le tableau 10 pour connaître les attributs facultatifs des objets métier enfants pour les informations relatives aux pièces jointes.

Tableau 10. Attributs facultatifs d'objets métier enfants

Nom	Description
AttachmentName	Nom de la pièce jointe. Cet attribut remplace les valeurs définies dans la propriété de l'extension de la pièce jointe. Il est au format nom de la pièce jointe + extension.

Tableau 10. Attributs facultatifs d'objets métier enfants (suite)

Nom	Description
DataEncoding	Indique si l'objet métier doit être converti en un pièce jointe binaire ou non binaire. Si cette valeur n'est pas définie, l'adaptateur utilisera la valeur DataEncoding indiquée dans les attributs de méta-objet de l'adaptateur.
MimeType	Instanciation MimeType pour DataHandler. Si cette valeur n'est pas définie, l'adaptateur utilisera la valeur MimeType définie dans les attributs de méta-objet de l'adaptateur.

En cas d'utilisation d'un objet métier enfant pour préciser les informations concernant la pièce jointe, le nom de l'attribut qui se mappe sur l'objet métier enfant doit être mentionné dans l'ASI de l'objet métier de premier niveau. Par exemple, `cw_mo_conn=<child_bo_attr_name>`.

Structure de Inbound Wrapper BO

La fonctionnalité Inbound Wrapper BO permet au connecteur d'assembler tous les objets métier des pièces jointes des messages électroniques et de les envoyer ensemble au courtier d'intégration. La structure InboundWrapperBO a également des attributs tels que FromAddress et ToAddress, renseignés par le connecteur. Voir le tableau 11 pour connaître les différents attributs pris en charge par la structure InboundWrapperBO.

Tableau 11. Attributs InboundWrapperBO pris en charge

Nom	Type	Cardinalité	Signification	Obligatoire dans la structure de l'objet métier
FromAddress	String	1	En-tête "From" du message électronique	Oui
ToAddress	String	1	En-tête "To" du message électronique	Oui
CC	String	1	En-tête "CC" du message électronique	Oui
BCC	String	1	En-tête "BCC" du message électronique	Oui
Subject	String	1	Objet du courrier	Oui
ReplyTo	String	1	En-tête ReplyTo du message électronique	Oui
SentDate	String	1	Date et heure d'envoi du message électronique	Oui

Tableau 11. Attributs InboundWrapperBO pris en charge (suite)

Nom	Type	Cardinalité	Signification	Obligatoire dans la structure de l'objet métier
ReceivedDate	String	1	Date et heure de réception du message électronique	Oui
<actual_bo>	<bo_type>	<1/N>	Type d'objet métier réel. Exemples : Customer Invoice PurchaseOrder	Oui

Remarque : Dans le tableau 11, <actual_bo> contient l'objet métier central (Customer, Invoice, etc.) créé à l'aide du gestionnaire de données. Chaque objet métier enfant a également un objet métier enfant pour conserver les informations liées à la pièce jointe. Comme ces valeurs d'attribut sont renseignées dynamiquement par le connecteur, le nom doit être indiqué dans l'ASI au niveau de l'objet métier, sous la forme suivante :

`cw_mo_conn=<child_bo_attribute_name>`

où `child_bo_attribute_name` est le nom de l'attribut qui se mappe sur le méta-objet de la pièce jointe. Pour obtenir une liste des attributs pris en charge par le méta-objet de la pièce jointe, voir le tableau 12.

Tableau 12. Attributs pris en charge par le méta-objet de la pièce jointe

Nom	Type	Cardinalité	Signification	Obligatoire dans la structure de l'objet métier
AttachmentName	String	1	Nom de la pièce jointe	Oui
Size*	String	1	Taille de la pièce jointe	Oui
MimeType	String	1	MimeType de la pièce jointe	Oui

Remarque : * Taille en octets. La taille ne se mappe pas sur la taille exacte de la pièce jointe, mais retourne une estimation utilisée pour calculer la taille du contenu affiché dans une interface utilisateur.

Attributs précisant les objets métier

Tout objet métier peut être inclus dans l'objet métier pour e-mail de premier niveau, ainsi longtemps que l'objet métier contenu est conforme aux exigences imposées par le gestionnaire de données configuré. Si un attribut est un objet métier, sa cardinalité doit être égale à 1.

Lors de la création de définitions d'objets métier pour l'adaptateur pour e-mail, gardez à l'esprit qu'un gestionnaire de données de l'adaptateur doit placer dans une unique pièce jointe chaque objet métier contenu dans l'objet métier de premier niveau. Par défaut, l'adaptateur convertit chaque pièce jointe en un objet métier et l'envoie séparément au courtier. Toutefois, si vous configurez la propriété `InboundWrapperBO`, l'adaptateur rassemblera tous les objets métier des pièces jointes de message du gestionnaire de données, et les enverra ensemble au courtier.

Pour cela, l'adaptateur créera un objet métier d'encapsuleur. L'objet métier d'encapsuleur contiendra les informations sur le courrier électronique, ainsi que les objets métier réels créés par le gestionnaire de données (tel que `Customer` ou `Invoice`). Ensuite, chaque objet métier inclus dans l'objet métier d'encapsuleur aura un objet métier enfant pour conserver les informations relatives aux pièces jointes. Pour plus d'informations sur les pièces jointes, voir «Structure facultative d'objet métier enfant pour la personnalisation des noms des pièces jointes et l'envoi de pièces jointes binaires», à la page 34.

La structure d'objet métier d'encapsuleur a des attributs (tels que `FromAddress`, `ToAddress` et `ReplyTo`) qui sont renseignés de façon dynamique par l'adaptateur quand la propriété `InboundWrapperBO` a été définie. Pour que l'adaptateur effectue cette action, le nom de l'objet métier enfant doit avoir été identifié dans l'ASI d'objet métier, avec le format suivant :

`cw_mo_conn=<child_bo_attr_name>`

`<child_bo_attr_name>` est le nom de l'attribut qui se mappe sur le méta-objet de la pièce jointe. Pour connaître les attributs pris en charge par le méta-objet de la pièce jointe, voir le tableau 13.

Tableau 13. Attributs pris en charge par le méta-objet de la pièce jointe

Nom	Type	Cardinalité	Signification	Obligatoire dans la structure de l'objet métier
AttachmentName	String	1	Nom de la pièce jointe	Oui
Size	String	1	Taille approximative de la pièce jointe (en octets)	Oui
MimeType	String	1	MimeType de la pièce jointe	Oui

Attributs précisant les fichiers

Pour joindre un fichier à un message électronique, indiquez le nom de fichier complet, y compris le nom du chemin. Si le fichier réside sur une autre machine, indiquez le nom du fichier en appliquant la convention de dénomination UNC. Par exemple, si le fichier réside sur une machine nommée Ernesto, indiquez le chemin de la façon suivante : `\\Ernesto\Email\EmailMessage.txt`. Vous pouvez également mapper un lecteur réseau sur la machine et indiquer le nom de fichier sous la forme `F:\Email\EmailMessage.txt`.

Mode de traitement des objets métier et des fichiers

Une fois que l'adaptateur a traité l'ensemble des attributs de base de l'objet métier de premier niveau, il recherche des attributs supplémentaires. Ils peuvent être de type String ou objet métier. Si l'attribut est d'un autre type, l'adaptateur l'ignore. L'adaptateur traite chaque attribut non null supplémentaire de la façon suivante :

- Si l'attribut est de type string, l'adaptateur le traite comme un nom de fichier complet. L'adaptateur attache le fichier en tant que pièce jointe au message électronique. Si le fichier n'est pas à l'emplacement précisé dans le nom de fichier, l'adaptateur retourne `BON_FAIL`.
- Si l'attribut est de type objet métier, l'adaptateur extrait l'attribut `MimeType` de l'objet métier de premier niveau, pour déterminer quel gestionnaire de données utiliser afin de convertir l'objet métier en chaîne. Si l'adaptateur ne parvient pas à déterminer le gestionnaire de données à utiliser d'après le type MIME, ou si l'attribut `MimeType` est null, l'adaptateur retourne `BON_FAIL`. Lorsqu'un gestionnaire de données a été instancié, l'adaptateur lui transmet l'objet métier et reçoit en retour une chaîne d'objet métier. L'adaptateur place la chaîne d'objet métier dans une pièce jointe du message électronique. Le nom de la pièce jointe est le nom de l'objet métier, et le type MIME est indiqué par l'attribut `MimeType` de l'objet métier de premier niveau.

Après avoir traité tous les attributs non null de l'objet métier de premier niveau, l'adaptateur envoie le message.

Conformité de l'objet métier avec les exigences du gestionnaire de données

Vous pouvez inclure tout objet métier dans l'objet métier d'encapsuleur de premier niveau de l'adaptateur pour e-mail, mais ces objets métier doivent livrer les données dans un format conforme aux exigences du gestionnaire de données utilisé pour convertir les données.

Par exemple, pour le gestionnaire de données `BySize`, une définition d'objet métier doit préciser la propriété d'attribut `MaxLength` pour chaque attribut d'objet métier. Pour le gestionnaire de données XML, la définition de l'objet métier doit inclure un texte spécifique à l'application lui permettant de générer un document XML.

Vous devez donc créer vos propres définitions d'objets métier pour chaque type de données à traiter. Dans la définition de l'objet métier, indiquez uniquement les données requises par l'application et les informations exigées par le gestionnaire de données. Vous pouvez ensuite inclure ces objets métier dans un objet métier de premier niveau d'adaptateur pour e-mail.

Voir le *Data Handler Guide* pour obtenir des informations spécifiques à chaque gestionnaire de données.

Traitement des instructions d'objet métier

Lors du traitement des requêtes d'objet métier, l'adaptateur pour e-mail gère uniquement l'instruction `Create`. Il indique un échec pour toute autre instruction. L'adaptateur conserve l'instruction des objets métier enfants.

Pour la notification des événements, chaque courrier électronique peut générer plusieurs objets métier. L'application qui envoie le message électronique est chargée de définir l'instruction de chaque objet métier. Le gestionnaire de données ne traite pas les instructions de ces objets métier, mais il les répercute dans les objets métier qu'il génère.

Propriétés des attributs des objets métier

Les attributs d'objet métier ont des propriétés qui peuvent affecter la façon dont l'adaptateur et le courtier d'intégration traitent ces attributs. Le tableau 14 indique comment l'adaptateur pour e-mail utilise ces propriétés pour les attributs de l'objet métier de premier niveau.

Tableau 14. Propriétés d'attribut d'objets métier

Propriété d'attribut	Description
Required	Pour les requêtes d'objet métier, l'adaptateur pour e-mail vérifie si la propriété Required est définie sur True.
Valeurs par défaut	Si l'objet métier ne fournit pas de valeur pour un attribut Required, et si une valeur par défaut est précisée, l'adaptateur utilise cette dernière.
Max Length	Non utilisé
Type	Non utilisé
Key	Non utilisé
Foreign Key	Non utilisé
Application-Specific Information	Non utilisé

Pour la notification d'événement, l'adaptateur pour e-mail n'utilise pas de propriétés d'attributs d'objet métier.

Meta-objets du gestionnaire de données

Si vous souhaitez que l'adaptateur pour e-mail puisse traiter des requêtes d'objet métier ou des notifications d'événement, vous devez définir un méta-objet de gestionnaire de données précisant quels gestionnaires de données l'adaptateur doit utiliser pour convertir les données.

Le méta-objet de premier niveau du gestionnaire de données est un objet métier hiérarchique qui contient n'importe quel nombre d'objets enfants. Chaque objet enfant est un objet métier plat représentant une instance de gestionnaire de données spécifique. Les méta-objets enfants ont des attributs qui fournissent des valeurs de configuration nécessaires au fonctionnement d'une instance de gestionnaire de données. Comme chaque type de gestionnaires de données exige des propriétés de configuration différentes, les méta-objets enfants qui prennent en charge des gestionnaires donnés ont des attributs différents.

Pour configurer un gestionnaire de données pour l'adaptateur pour e-mail, procédez comme suit :

- Paramétrez le meta-objet du gestionnaire de données de premier niveau pour avoir un attribut pour chaque type MIME qui doit être pris en charge par l'adaptateur. Le nom de l'attribut doit être le nom du type MIME. L'attribut représente un méta-objet enfant pour une instance de gestionnaire de données. Le méta-objet généralement utilisé pour les adaptateurs est nommé `MO_DataHandler_Default`.

Notez que pour les pièces jointes reçues dans des notifications d'événement, l'adaptateur pour e-mail convertit le type MIME de la pièce jointe en minuscules avant d'instancier le gestionnaire de données. Par conséquent, lorsque vous indiquez le type MIME dans le méta-objet du gestionnaire de données, assurez-vous qu'il soit en minuscules. Dans le cas contraire, l'adaptateur ne peut pas instancier le gestionnaire de données.

- Définissez les valeurs d'attribut par défaut pour chaque méta-objet enfant. Les attributs des gestionnaires de données fournis par IBM sont décrits dans le *Data Handler Guide*.

Pour des informations détaillées sur la configuration des méta-objets pour chaque gestionnaire de données, voir le *Data Handler Guide*.

Remarque : Pour que l'adaptateur pour e-mail puisse instancier un gestionnaire de données, le méta-objet de premier niveau du gestionnaire de données doit être configuré dans la liste des objets métier pris en charge par l'adaptateur.

Utilisation des méta-objets de l'adaptateur pour e-mail

Les méta-objets sont des objets métier conçus pour contenir les informations de configuration des adaptateurs. Un méta-objet est nécessaire pour configurer l'adaptateur pour e-mail pour la notification d'événement.

L'adaptateur pour e-mail extrait le nom du méta-objet de premier niveau depuis la propriété PollConfigMO de l'adaptateur. Vous devez créer la définition de ce méta-objet en respectant la structure décrite dans «Structure du méta-objet de l'adaptateur pour e-mail», à la page 40. Indiquez ensuite son nom dans la propriété PollConfigMO. Pour plus d'informations sur la création d'une définition d'objet métier, voir *Business Object Development Guide*.

Important : Pour que l'adaptateur de messagerie puisse utiliser les informations de configuration du méta-objet, vous devez créer la définition du méta-objet et l'ajouter à la liste des objets métier pris en charge par l'adaptateur.

Structure du méta-objet de l'adaptateur pour e-mail

Le méta-objet de l'adaptateur pour e-mail est un objet métier hiérarchique dont l'objet de premier niveau peut contenir n'importe quel nombre d'objets enfants. Chaque objet enfant représente un emplacement d'interrogation. Pour configurer l'adaptateur de façon à contrôler un ou plusieurs emplacements d'interrogation, vous devez paramétrer le méta-objet avec un nombre correspondant de méta-objets enfants. La figure 5, à la page 43 illustre la définition d'un méta-objet.

Attributs de méta-objet de premier niveau

Tous les attributs du méta-objet de premier niveau sont des attributs de conteneur dont la cardinalité est égale à 1. Par exemple, si l'adaptateur a deux emplacements d'interrogation, Location1 et Location2, les types d'attributs peuvent être MO_PollLocation1 et MO_PollLocation2.

Attributs de méta-objets enfants

Un méta-objet enfant est un objet plat doté des attributs indiqués dans le tableau 15.

Tableau 15. Attributs de méta-objets enfants

Nom	Description
PollHostName	Nom de la machine hôte qui héberge le serveur de messagerie. La boîte aux lettres est située sur ce serveur. JavaMail doit connaître ce nom pour se connecter au magasin de messages.

Tableau 15. Attributs de méta-objets enfants (suite)

Nom	Description
UserName	Nom de l'utilisateur. L'adaptateur interroge la boîte aux lettres de cet utilisateur. JavaMail doit connaître ce nom pour authentifier l'utilisateur.
Password	Le mot de passe de l'utilisateur. JavaMail doit connaître cette valeur pour authentifier l'utilisateur.
PollFolder	Nom du dossier dans la boîte aux lettres. L'adaptateur interroge ce dossier. Ce dossier doit exister dans la boîte aux lettres de l'utilisateur. La valeur par défaut est PollFolder. BiDi pris en charge par Windows. Toutes les propriétés conformes BiDi seront converties depuis le format Windows BiDi dans le format BiDi.Metadata, qui est une propriété standard de connecteur E-mail.
ArchiveFolder	Nom du dossier dans la boîte aux lettres. L'adaptateur archive les messages qui ont abouti (événements réussis) dans ce dossier. Ce dossier doit exister dans la boîte aux lettres de l'utilisateur. Si aucune valeur n'est indiquée pour cet attribut, l'adaptateur supprime les messages qui ont abouti après les avoir traités. BiDi pris en charge par Windows. Toutes les propriétés conformes BiDi seront converties depuis le format Windows BiDi dans le format BiDi.Metadata, qui est une propriété standard de connecteur E-mail.
FailFolder	Nom du dossier dans la boîte aux lettres. L'adaptateur archive dans ce dossier les messages qui n'ont pas abouti (événements d'échec). Si aucune valeur n'est indiquée pour cet attribut, l'adaptateur supprime les messages qui ont échoué. BiDi pris en charge par Windows. Toutes les propriétés conformes BiDi seront converties depuis le format Windows BiDi dans le format BiDi.Metadata, qui est une propriété standard de connecteur E-mail.
ExternalProperty	Chemin absolu du nom du fichier de propriété utilisé lorsque l'adaptateur se connecte au serveur de messagerie pour l'interrogation. Exemple : mail.imap.partialfetch=false

Tableau 15. Attributs de méta-objets enfants (suite)

Nom	Description
InDoubtEvents	Les valeurs valides pour cet attribut sont <code>Reprocess</code> , <code>FailOnStartup</code> , <code>LogError</code> et <code>Ignore</code> . En cas d'événements en attente de validation dans cet emplacement d'interrogation, l'adaptateur les traite au cours de l'initialisation en fonction de la valeur par défaut définie pour cet attribut. Si la valeur est définie sur <code>Reprocess</code> , l'adaptateur réinitialise l'indicateur SEEN des événements en attente de validation dans le dossier d'interrogation, et prélève ces messages dans les appels d'interrogation suivants. Si la valeur est définie sur <code>FailOnStartup</code> et s'il y a des transactions en attente de validation dans ce dossier d'interrogation, l'adaptateur échoue. Si la valeur est définie sur <code>LogError</code> et s'il y a des transactions en attente de validation dans ce dossier d'interrogation, l'adaptateur consigne l'erreur et continue. Si la valeur est définie sur <code>Ignore</code> et s'il y a des transactions en attente de validation dans ce dossier d'interrogation, l'adaptateur les ignore. La valeur par défaut est <code>Reprocess</code> .
Reconnect	Les valeurs possibles pour cet attribut sont <code>True</code> ou <code>False</code> . Cette valeur est utilisée pour déterminer si une tentative de reconnexion doit être effectuée pour chaque cycle d'interrogation, en cas de perte de connexion.

Exemple de méta-objet

La figure 5 présente un exemple de méta-objet de premier niveau nommé `MO_Email_Default`. L'adaptateur pour e-mail utilise ce méta-objet pour interroger deux emplacements, indiqués dans les deux méta-objets enfants `MO_PollLocation1` et `MO_PollLocation2`.

Dans l'exemple, la valeur de l'attribut `EventRecovery` de `MO_PollLocation1` est `Resubmit`, et la valeur de cet attribut dans `MO_PollLocation2` est `Ignore`. La valeur de l'attribut `EventRecovery` indique à l'administrateur de l'adaptateur de soumettre de nouveau les événements en attente de validation et de les ignorer pour `Location2`.

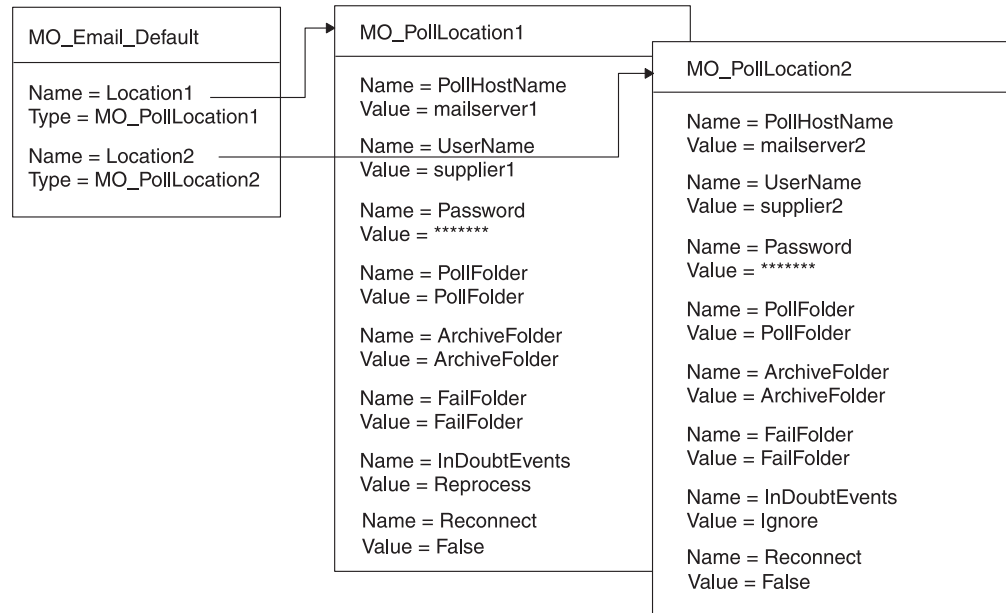


Figure 5. Exemple de méta-objet pour l'adaptateur pour e-mail

Objet métier EmailNotification

Remarque : EmailNotification Business Object (uniquement pour le courtier d'intégration InterChange Server)

Si votre courtier d'intégration est InterChange Server, alors un objet métier nommé EmailNotification est automatiquement créé par le système pour prendre en charge la notification des erreurs. Lorsqu'une erreur se produit, l'objet métier EmailNotification est instancié par le système et envoyé à l'adaptateur pour e-mail. L'adaptateur extrait les informations du message depuis l'objet métier, crée le message électronique et l'envoie.

L'objet métier EmailNotification contient les attributs indiqués dans le tableau 16. La seule instruction prise en charge pour cet objet métier est Create.

Tableau 16. attributs d'objet métier EmailNotification

Nom	Description	Obligatoire ?
RecipientName	Destinataire du message. Pour plus d'informations, voir «Indication d'adresses e-mail valides», à la page 45.	Oui
MessageSubject	Description du contenu du message électronique. Valeur définie dans l'objet métier par la collaboration.	Oui. Si aucune valeur n'est définie, l'adaptateur utilise la valeur par défaut.
MessageContent	Contenu du message électronique. Valeur définie dans l'objet métier par la collaboration.	Oui. Si aucune valeur n'est définie, l'adaptateur utilise la valeur par défaut.

Tableau 16. attributs d'objet métier EmailNotification (suite)

Nom	Description	Obligatoire ?
MessageAttachment	Fichier à joindre au message.	Facultatif. Si aucune valeur n'est définie, aucune pièce n'est jointe au message. Pour joindre un fichier à un message électronique, indiquez le nom de fichier complet, y compris le chemin.
FromAddress	Valeurs définies dans l'objet métier lors de la configuration. Pour plus d'informations, voir «Indication d'adresses e-mail valides», à la page 45.	Oui
BusinessObjectMimeCharset	Le jeu de caractères des objets métier contenus dans l'e-mail. Pour plus d'informations sur cette propriété, voir sa description dans la définition de l'adaptateur, dans la section «BusinessObjectMimeCharset», à la page 21.	Non
BusinessObjectMimeEncoding	Le codage des objets métier contenus dans l'e-mail. Pour plus d'informations sur cette propriété, voir sa description dans la définition de l'adaptateur, dans la section «BusinessObjectMimeEncoding», à la page 21.	Non
MessageContentMimeCharset	Le jeu de caractères du contenu du message. Pour plus d'informations sur cette propriété, voir sa description dans la définition de l'adaptateur, dans la section «MessageContentMimeCharset», à la page 22.	Non
MessageContentMimeEncoding	Le codage du contenu du message. Pour plus d'informations sur cette propriété, voir sa description dans la définition de l'adaptateur, dans la section «MessageContentMimeEncoding», à la page 22.	Non
MessageHeaderMimeCharset	Le jeu de caractère de l'en-tête du message. Pour plus d'informations sur cette propriété, voir sa description dans la définition de l'adaptateur, dans la section «MessageHeaderMimeCharset», à la page 22.	Non
MessageHeaderMimeEncoding	Le codage de l'en-tête du message. Pour plus d'informations sur cette propriété, voir sa description dans la définition de l'adaptateur, dans la section «MessageHeaderMimeEncoding», à la page 22.	Non
ReplyTo	Adresse(s) ReplyTo du message électronique Insère la valeur de l'attribut MessageContent dans la zone CONTENT. Insère la valeur de l'attribut MessageAttachment comme pièce jointe au message. Insère la valeur de l'attribut FromAddress dans la zone FROM. Insère la valeur de l'attribut ReplyTo dans la zone REPLYTO.	Non

Remarque : Aucun message électronique n'est envoyé si un nom de fichier de pièce jointe est incorrect, si le fichier n'existe pas ou si la pièce jointe n'est pas lisible. Si le fichier réside sur une autre machine, indiquez le nom de fichier en appliquant la convention de dénomination UNC. Par exemple, si le fichier réside sur une machine nommée Ernesto, indiquez le chemin de la façon suivante :
\\Ernesto\Email\EmailMessage. Vous pouvez également mapper un lecteur réseau sur la machine et indiquer le nom de fichier, sous la forme F:\Email\EmailMessage.

Lors de l'exécution, l'adaptateur extrait les valeurs d'attribut de l'objet métier et les insère dans un message électronique, de la façon suivante :

- Insertion dans la zone TO de un ou plusieurs destinataires du message, de la façon définie dans l'attribut RecipientName.
- Insertion de la valeur de l'attribut MessageSubject dans la zone SUBJECT.
- Insertion de la valeur de l'attribut MessageContent dans la zone CONTENT.
- Insertion de la valeur de l'attribut MessageAttachment comme pièce jointe au message.
- Insertion de la valeur de l'attribut FromAddress dans la zone FROM.

Indication d'adresses e-mail valides

Une adresse valide peut être constituée d'une ou plusieurs adresses Internet complètes séparées par des virgules, comme indiqué par la RFC 822. Voici un exemple de saisie valide pour deux destinataires :

```
JohnDoe@company.com,"/FredSmith/" <FredSmith@company.com>
```

L'adaptateur ne convertit pas les alias d'adresse personnelle, tels que l'alias EngineeringDepartment défini dans un carnet d'adresses personnel. Toutefois, une adresse valide peut être un alias défini dans un serveur de messagerie, comme EngineeringDepartment ou EngineeringDepartment@company.com. Dans ce cas, le serveur de messagerie décode l'alias et envoie l'e-mail à tous ses membres.

L'adaptateur pour e-mail procède à l'analyse syntaxique de chaque adresse individuelle pour contrôler si elle est conforme à la RFC 822. Voici des exemples d'adresses valides :

- name@company.com
- EngineeringDepartment ou EngineeringDepartment@company.com, où EngineeringDepartment est un alias défini sur le serveur de messagerie. Le serveur de messagerie décode l'alias et envoie le message à tous ses membres.
- person1@some_company.com, person2@another_company.com. Vous pouvez entrer plusieurs adresses, pourvu qu'elles soient séparées par des virgules. Par exemple, "NAME1" <name1@company.com>, <name2@company.com> .

Si une adresse électronique exige un signe at (@), vous devez l'indiquer dans l'adresse puisque l'adaptateur n'ajoute pas lui-même de caractère à une adresse.

Annexe A. Propriétés standard de configuration pour les adaptateurs

Cette annexe décrit les propriétés de configuration standard pour le composant de connecteur de WebSphere Business Integration Adapters. Les informations présentées concernent les connecteurs qui s'exécutent sur les courtiers d'intégration suivants :

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator, WebSphere MQ Integrator Broker et WebSphere Business Integration Message Broker, appelés collectivement courtiers de messages WebSphere (et WMQI dans Connector Configurator).
- Information Integrator (II)
- WebSphere Application Server (WAS)

Si votre adaptateur prend en charge DB2 Information Integrator, utilisez les options et les propriétés standard DB2 II (voir la colonne Remarque du tableau 17, à la page 49.)

Les propriétés définies pour l'adaptateur dépendent du courtier d'intégration utilisé. Vous sélectionnez ce dernier à l'aide de Connector Configurator. Une fois le courtier choisi, Connector Configurator dresse la liste des propriétés standard à configurer pour l'adaptateur.

Pour plus d'informations sur les propriétés spécifiques au connecteur, voir la section correspondante de ce guide.

Nouvelles propriétés

La propriété standard suivante a été ajoutée à cette édition :

- BOTrace

Présentation des propriétés de connecteur standard

Les connecteurs ont deux types de propriétés de configuration :

- Les propriétés de configuration standard, utilisées par l'architecture
- Les propriétés de configuration spécifiques à une application ou à un connecteur, et utilisées par l'agent

Ces propriétés déterminent l'architecture de l'adaptateur et le comportement d'exécution de l'agent.

Cette section indique comment démarrer Connector Configurator et décrit les caractéristiques communes à toutes les propriétés. Pour plus d'informations sur les propriétés de configuration spécifiques à un connecteur, reportez-vous au guide d'utilisateur de l'adaptateur approprié.

Démarrage de Connector Configurator

Vous pouvez configurer les propriétés du connecteur à partir de Connector Configurator, accessible via System Manager. Pour plus d'informations sur l'utilisation de Connector Configurator, voir les sections associées de ce guide.

Connector Configurator et System Manager s'exécutent uniquement sous Windows. Si vous exécutez le connecteur sous UNIX, vous devez posséder une machine Windows sur laquelle ces outils sont installés.

Pour définir les propriétés d'un connecteur s'exécutant sous UNIX, vous devez démarrer System Manager sur la machine Windows, établir une connexion au courtier d'intégration UNIX et mettre à jour Connector Configurator pour le connecteur.

Présentation des valeurs des propriétés de configuration

Le connecteur utilise l'ordre suivant pour déterminer la valeur d'une propriété :

1. Valeur par défaut
2. Référentiel (valide uniquement si WebSphere InterChange Server (ICS) est le courtier d'intégration)
3. Fichier de configuration locale
4. Ligne de commande

La longueur par défaut d'une propriété est de 255 caractères. La longueur d'un type de propriété STRING n'est pas limitée. La longueur d'un type INTEGER est déterminée par le serveur sur lequel l'adaptateur fonctionne.

Un connecteur obtient ses valeurs de configuration lors du démarrage. Si vous modifiez la valeur d'une ou plusieurs propriétés du connecteur pendant une session d'exécution, la méthode de mise à jour de la propriété détermine la manière dont les modifications prennent effet.

Les caractéristiques de mise à jour d'une propriété, c'est-à-dire à quel moment et de quelle façon la modification des propriétés du connecteur prend effet, dépendent de la nature de la propriété.

Il existe quatre méthodes de mise à jour pour les propriétés standard du connecteur :

- **Dynamique**
La nouvelle valeur prend effet dès que la modification est enregistrée dans System Manager. Toutefois, si le connecteur est en mode autonome (indépendamment de System Manager), par exemple avec l'un des courtiers de message WebSphere, vous ne pouvez modifier les propriétés que via le fichier de configuration. Dans ce cas, une mise à jour dynamique n'est pas possible.
- **Redémarrage de l'agent (ICS uniquement)**
La nouvelle valeur ne prend effet qu'une fois que vous avez arrêté et redémarré l'agent du connecteur.
- **Redémarrage du composant**
La nouvelle valeur ne prend effet qu'après que le connecteur ait été arrêté et redémarré dans System Manager. Vous n'avez pas besoin d'arrêter et de redémarrer l'agent ni le processus du serveur.
- **Redémarrage du système**
La nouvelle valeur ne prend effet qu'une fois que vous avez arrêté et redémarré l'agent du connecteur et le serveur.

Pour déterminer la manière dont une propriété donnée est mise à jour, reportez-vous à la colonne **Update Method** dans la fenêtre Connector Configurator, ou à la colonne Update Method dans le tableau 17, à la page 49.

Une propriété standard peut figurer à trois endroits. Certaines propriétés peuvent figurer à plusieurs emplacements.

- **ReposController**
La propriété réside dans le contrôleur du connecteur et n'est effective qu'à cet emplacement. Le fait de modifier la valeur du côté de l'agent n'a pas d'effet sur le contrôleur.
- **ReposAgent**
La propriété réside dans l'agent et n'est effective qu'à cet emplacement. Selon la propriété, une configuration locale peut remplacer cette valeur.
- **LocalConfig**
La propriété réside dans le fichier de configuration du connecteur et n'agit que par l'intermédiaire de ce fichier. Le contrôleur ne peut pas modifier la valeur de la propriété et n'est pas informé des modifications apportées au fichier de configuration, à moins que le système ne soit redéployé pour remettre à jour le contrôleur explicitement.

Référence rapide des propriétés standard

Le tableau 17 fournit une description rapide des propriétés standard de configuration des connecteurs. Les connecteurs n'exigent pas toutes ces propriétés, et les paramètres de propriété peuvent différer d'un courtier d'intégration à l'autre.

Voir la section qui suit le tableau pour une description de chaque propriété.

Remarque : Dans la colonne "Remarques" du tableau 17, la phrase "La valeur de RepositoryDirectory est égale à <REMOTE>" indique que le courtier est InterChange Server. Lorsque le courtier est WMQI ou WAS, le répertoire du référentiel est défini sur <ProductDir>\repository

Tableau 17. Récapitulatif des propriétés de configuration standard

Nom de propriété	Valeurs possibles	Valeur par défaut	Méthode de mise à jour	Remarques
AdapterHelpName	Un des sous-répertoires valides de <ProductDir>\bin\Data\App\Help\ contenant un répertoire <RegionalSetting> valide	Nom du modèle, si valide, ou zone vide	Redémarrage du composant	Paramètres régionaux pris en charge. Incluent chs_chn, cht_twn, deu_deu, esn_esp, fra_fra, ita_ita, jpn_jpn, kor_kor, ptb_bra, et enu_usa (par défaut).
AdminInQueue	Nom de file d'attente JMS valide	<CONNECTORNAME>/ADMININQUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS
AdminOutQueue	Nom de file d'attente JMS valide	<CONNECTORNAME>/ADMINOUTQUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS
AgentConnections	1 à 4	1	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est MQ ou IDL, la valeur de Repository Directory est <REMOTE> et la valeur de BrokerType est ICS.

Tableau 17. Récapitulatif des propriétés de configuration standard (suite)

Nom de propriété	Valeurs possibles	Valeur par défaut	Méthode de mise à jour	Remarques
AgentTraceLevel	0 à 5	0	Dynamique si le courtier est ICS ; sinon Redémarrage du composant	
ApplicationName	Nom d'application	Valeur précisée pour le nom de l'application du connecteur	Redémarrage du composant	
BiDi.Application	Toute combinaison valide de ces attributs bidirectionnels : 1ère lettre : I, V 2e lettre : L, R 3e lettre : Y, N 4e lettre : S, N 5e lettre : H, C, N	ILYNN (cinq lettres)	Redémarrage du composant	Cette propriété n'est valide que si la valeur de BiDi.Transformation est true
BiDi.Broker	Toute combinaison valide de ces attributs bidirectionnels : 1ère lettre : I, V 2e lettre : L, R 3e lettre : Y, N 4e lettre : S, N 5e lettre : H, C, N	ILYNN (cinq lettres)	Redémarrage du composant	Cette propriété n'est valide que si la valeur de BiDi.Transformation est true. Si la valeur de BrokerType est ICS, la propriété est en lecture seule.
BiDi.Metadata	Toute combinaison valide de ces attributs bidirectionnels : 1ère lettre : I, V 2e lettre : L, R 3e lettre : Y, N 4e lettre : S, N 5e lettre : H, C, N	ILYNN (cinq lettres)	Redémarrage du composant	Cette propriété n'est valide que si la valeur de BiDi.Transformation est true.
BiDi.Transformation	true ou false	false	Redémarrage du composant	Cette propriété n'est valide que si la valeur de BrokerType n'est pas WAS.
BOTrace	none ou keys ou full	none	Redémarrage du composant	Cette propriété n'est valide que si la valeur de AgentTraceLevel est inférieure à 5.
BrokerType	ICS, WMQI, WAS	ICS	Redémarrage du composant	
CharacterEncoding	Tout code pris en charge. La liste indique le sous-ensemble : ascii7, ascii8, SJIS, Cp949, GBK, Big5, Cp297, Cp273, Cp280, Cp284, Cp037, Cp437	ascii7	Redémarrage du composant	Cette propriété n'est valide que pour les connecteurs C++.
CommonEventInfrastructure	true ou false	false	Redémarrage du composant	
CommonEventInfrastructureURL	Une chaîne URL, par exemple, corbaloc:iiop:host:2809.	Aucune valeur par défaut.	Redémarrage du composant	Cette propriété n'est valide que si la valeur de CommonEventInfrastructure est true.

Tableau 17. Récapitulatif des propriétés de configuration standard (suite)

Nom de propriété	Valeurs possibles	Valeur par défaut	Méthode de mise à jour	Remarques
ConcurrentEventTriggeredFlows	1 à 32,767	1	Redémarrage du composant	Cette propriété n'est valide que si la valeur de RepositoryDirectory est <REMOTE> et la valeur de BrokerType est ICS.
ContainerManagedEvents	Vide ou JMS	Vide	Redémarrage du composant	Cette propriété n'est valide que si la valeur de Delivery Transport est JMS.
ControllerEventSequencing	true ou false	true	Dynamique	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
ControllerStoreAndForwardMode	true ou false	true	Dynamique	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
ControllerTraceLevel	0 à 5	0	Dynamique	Cette propriété n'est valide que si la valeur de RepositoryDirectory est <REMOTE> et la valeur de BrokerType est ICS.
DeliveryQueue	Tout nom valide de file d'attente JMS valide	<CONNECTORNAME>/DELIVERYQUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de Delivery Transport est JMS.
DeliveryTransport	MQ, IDL ou JMS	IDL lorsque la valeur de RepositoryDirectory est <REMOTE>, sinon JMS	Redémarrage du composant	Si la valeur de RepositoryDirectory n'est pas <REMOTE>, la seule valeur valide pour cette propriété est JMS.
DuplicateEventElimination	true ou false	false	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
EnableOidForFlowMonitoring	true ou false	false	Redémarrage du composant	Cette propriété n'est valide que si la valeur de BrokerType est ICS.
FaultQueue	Tout nom de file d'attente valide.	<CONNECTORNAME>/FAULTQUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory, CxCommon.Messaging.jms.SonicMQFactory, ou n'importe quel nom de classe Java	CxCommon.Messaging.jms.IBMMQSeriesFactory	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
jms.ListenerConcurrency	1 à 32767	1	Redémarrage du composant	Cette propriété n'est valide que si la valeur de jms.TransportOptimized est true.

Tableau 17. Récapitulatif des propriétés de configuration standard (suite)

Nom de propriété	Valeurs possibles	Valeur par défaut	Méthode de mise à jour	Remarques
jms.MessageBrokerName	Si la valeur de jms.FactoryClassName est IBM, utilisez crossworlds.queue.manager.	crossworlds.queue.manager	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
jms.NumConcurrentRequests	Entier positif	10	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
jms.Password	Tout mot de passe valide		Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
jms.TransportOptimized	true ou false	false	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS et la valeur de BrokerType est ICS.
jms.UserName	Tout nom valide		Redémarrage du composant	Cette propriété n'est valide que si la valeur de Delivery Transport est JMS.
JvmMaxHeapSize	Taille de segment en mégaoctets	128 Mo	Redémarrage du composant	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
JvmMaxNativeStackSize	Taille de la pile en kilo-octets	128 Ko	Redémarrage du composant	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
JvmMinHeapSize	Taille de segment en mégaoctets	1 Mo	Redémarrage du composant	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
ListenerConcurrency	1 à 100	1	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est MQ.
Locale	Il s'agit d'un sous-ensemble des paramètres régionaux pris en charge : en_US, ja_JP, ko_KR, zh_CN, zh_TW, fr_FR, de_DE, it_IT, es_ES, pt_BR	en_US	Redémarrage du composant	
LogAtInterchangeEnd	true ou false	false	Redémarrage du composant	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.

Tableau 17. Récapitulatif des propriétés de configuration standard (suite)

Nom de propriété	Valeurs possibles	Valeur par défaut	Méthode de mise à jour	Remarques
MaxEventCapacity	1 à 2147483647	2147483647	Dynamique	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
MessageFileName	Nom de fichier valide	InterchangeSystem.txt	Redémarrage du composant	
MonitorQueue	Tout nom de file d'attente valide	<CONNECTORNAME>/MONITORQUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DuplicateEventElimination est true et si ContainerManagedEvents n'a pas de valeur.
OADAutoRestartAgent	true ou false	false	Dynamique	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
OADMaxNumRetry	Un nombre entier positif	1000	Dynamique	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
OADRetryTimeInterval	Un nombre entier positif en minutes	10	Dynamique	Cette propriété n'est valide que si la valeur de Repository Directory est égale à <REMOTE> et la valeur de BrokerType est ICS.
PollEndTime	HH = 0 à 23 MM = 0 à 59	HH:MM	Redémarrage du composant	
PollFrequency	Un nombre entier positif (en millisecondes)	10000	Dynamique si le courtier est ICS ; sinon Redémarrage du composant	
PollQuantity	1 à 500	1	Redémarrage de l'agent	Cette propriété n'est valide que si la valeur de ContainerManagedEvents est JMS.
PollStartTime	HH = 0 à 23 MM = 0 à 59	HH:MM	Redémarrage du composant	
RepositoryDirectory	<REMOTE> si le courtier est ICS ; sinon tout répertoire local valide.	Pour ICS, la valeur est définie sur <REMOTE> Pour WMQI et WAS, la valeur est <ProductDir\repository	Redémarrage de l'agent	
RequestQueue	Nom de file d'attente JMS valide	<CONNECTORNAME>/REQUESTQUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS

Tableau 17. Récapitulatif des propriétés de configuration standard (suite)

Nom de propriété	Valeurs possibles	Valeur par défaut	Méthode de mise à jour	Remarques
ResponseQueue	Nom de file d'attente JMS valide	<CONNECTORNAME>/RESPONSEQUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
RestartRetryCount	0 à 99	7	Dynamique si ICS ; sinon Redémarrage du composant	
RestartRetryInterval	Une valeur en minutes de 1 à 2147483647	1	Dynamique si ICS ; sinon Redémarrage du composant	
ResultsSetEnabled	true ou false	false	Redémarrage du composant	Utilisé uniquement par les connecteurs qui prennent en charge DB2II. Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS, et la valeur de BrokerType est WMQI.
ResultsSetSize	Entier positif	0 (indique que la taille de l'ensemble de résultats est illimitée)	Redémarrage du composant	Utilisé uniquement par les connecteurs qui prennent en charge DB2II. Cette propriété n'est valide que si la valeur de ResultsSetEnabled est true.
RHF2MessageDomain	mrm ou xml	mrm	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS et la valeur de WireFormat est CwXML.
SourceQueue	Tout nom de file d'attente WebSphere MQ	<CONNECTORNAME>/SOURCEQUEUE	Redémarrage de l'agent	Cette propriété n'est valide que si la valeur de ContainerManagedEvents est JMS.
SynchronousRequest Queue	Tout nom de file d'attente valide.	<CONNECTORNAME>/SYNCHRONOUSREQUEST QUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
SynchronousRequest Timeout	0 à n'importe quelle valeur (millisecondes)	0	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
SynchronousResponse Queue	Tout nom de file d'attente valide	<CONNECTORNAME>/SYNCHRONOUSRESPONSE QUEUE	Redémarrage du composant	Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS.
TivoliMonitorTransaction Performance	true ou false	false	Redémarrage du composant	

Tableau 17. Récapitulatif des propriétés de configuration standard (suite)

Nom de propriété	Valeurs possibles	Valeur par défaut	Méthode de mise à jour	Remarques
WireFormat	CwXML ou CwBO	CwXML	Redémarrage de l'agent	La valeur de cette propriété doit être CwXML si la valeur de RepositoryDirectory n'est pas définie sur <REMOTE>. La valeur doit être CwBO si la valeur de RepositoryDirectory est définie sur <REMOTE>.
WsifSynchronousRequestTimeout	0 à n'importe quelle valeur (millisecondes)	0	Redémarrage du composant	Cette propriété n'est valide que si la valeur de BrokerType est WAS.
XMLNamespaceFormat	short ou long ou no	short	Redémarrage de l'agent	Cette propriété n'est valide que si la valeur de BrokerType est WMQI ou WAS

Propriétés standard

Cette section décrit les propriétés standard de configuration du connecteur.

AdapterHelpName

La propriété AdapterHelpName est le nom d'un répertoire contenant des fichiers d'aide étendue spécifiques au connecteur. Le répertoire doit figurer dans <ProductDir>\bin\Data\App\Help et contenir au moins le répertoire de langue enu_usa. Il peut contenir d'autres répertoires selon les paramètres régionaux.

La valeur par défaut est le nom du modèle s'il est valide, ou elle est vide.

AdminInQueue

La propriété AdminInQueue précise la file d'attente utilisée par le courtier d'intégration pour envoyer des messages administratifs au connecteur.

La valeur par défaut est <CONNECTORNAME>/ADMININQUEUE

AdminOutQueue

La propriété AdminOutQueue précise la file d'attente utilisée par le connecteur pour envoyer des messages administratifs au courtier d'intégration.

La valeur par défaut est <CONNECTORNAME>/ADMINOUTQUEUE

AgentConnections

La propriété AgentConnections contrôle le nombre de connexions ORB (Object Request Broker) ouvertes à l'initialisation de ORB.

Elle n'est valide que si la valeur de RepositoryDirectory est définie sur <REMOTE> et si la valeur de la propriété DeliveryTransport est MQ ou IDL.

La valeur par défaut de cette propriété est 1.

AgentTraceLevel

La propriété AgentTraceLevel définit le niveau des messages de trace pour le composant spécifique à l'application. Le connecteur fournit tous les messages de trace applicables au niveau de trace défini et à un niveau inférieur.

La valeur par défaut est 0.

ApplicationName

La propriété ApplicationName identifie de façon unique le nom de l'application du connecteur. Ce nom permet à l'administrateur système de surveiller l'environnement d'intégration. Vous devez attribuer une valeur à cette propriété avant d'exécuter le connecteur.

La valeur par défaut est le nom du connecteur.

BiDi.Application

La propriété BiDi.Application précise le format bidirectionnel des données provenant d'une application externe et entrant dans l'adaptateur, sous la forme d'un objet métier pris en charge par cet adaptateur. La propriété définit les attributs bidirectionnels des données de l'application. Ces attributs sont les suivants :

- Type de texte : implicite ou visuel (I ou V)
- Direction du texte : de gauche à droite ou de droite à gauche (L ou R)
- Permutation symétrique : activée ou désactivée (Y ou N)
- Mise en forme (arabe): activée ou désactivée (S ou N)
- Mise en forme numérique (arabe) : hindi, contextuel, ou nominal (H, C ou N)

Cette propriété n'est valide que si la valeur de la propriété BiDi.Transformation est définie sur true.

La valeur par défaut est ILYNN (implicite, gauche à droite, activé, désactivé, nominal).

BiDi.Broker

La propriété BiDi.Broker précise le format de script bidirectionnel pour les données envoyées depuis l'adaptateur au courtier d'intégration sous la forme d'un objet métier pris en charge. Elle définit les attributs bidirectionnels des données, indiqués sous BiDi.Application ci-dessous.

Cette propriété n'est valide que si la valeur de la propriété BiDi.Transformation est définie sur true. Si la propriété BrokerType est ICS, sa valeur est en lecture seule.

La valeur par défaut est ILYNN (implicite, gauche à droite, activé, désactivé, nominal).

BiDi.Metadata

La propriété BiDi.Metadata définit le format bidirectionnel ou les attributs des métadonnées qui sont utilisées par le connecteur pour établir et maintenir un lien vers l'application externe. Les paramètres de l'attribut sont spécifiques à chaque adaptateur qui utilise des capacités bidirectionnelles. Si votre adaptateur prend en charge le traitement bidirectionnel, voir la section relative aux propriétés spécifiques à l'adaptateur pour plus d'informations.

Cette propriété n'est valide que si la valeur de la propriété BiDi.Transformation est définie sur true.

La valeur par défaut est ILYNN (implicite, gauche à droite, activé, désactivé, nominal).

BiDi.Transformation

La propriété BiDi.Transformation détermine si le système procède ou non à une transformation bidirectionnelle lors de l'exécution.

Si la valeur de la propriété est définie sur true, les propriétés BiDi.Application, BiDi.Broker et BiDi.Metadata sont disponibles. Si la valeur de la propriété est définie sur false, elles sont cachées.

La valeur par défaut est false.

BOTrace

La propriété BOTrace indique si les messages de trace d'objet métier sont activés ou non lors de l'exécution.

Remarque : Ceci ne s'applique que si la propriété AgentTraceLevel est inférieure à 5.

Lorsque le niveau de trace est inférieur à 5, vous pouvez utiliser ces paramètres de ligne de commande pour réinitialiser la valeur de BOTrace.

- Entrez -xBOTrace=Full pour afficher tous les attributs d'objets métier.
- Entrez -xBOTrace=Keys pour n'afficher que les clés d'objets métier.
- Entrez -xBOTrace=None pour désactiver l'affichage des attributs d'objets métier.

La valeur par défaut est false.

BrokerType

La propriété BrokerType identifie le type de courtier d'intégration que vous utilisez. Les valeurs possibles sont ICS, WMQI (pour WMQI, WMQIB ou WBIMB) ou WAS.

CharacterEncoding

La propriété CharacterEncoding indique le jeu de codes de caractères utilisé pour mettre en correspondance un caractère (une lettre de l'alphabet, un chiffre ou un signe de ponctuation) et une valeur numérique.

Remarque : Les connecteurs Java n'utilisent pas cette propriété. Les connecteurs C++ utilisent la valeur ascii7 pour cette propriété.

Par défaut, n'est affiché qu'un sous-ensemble des codages de caractères pris en charge. Pour ajouter d'autres valeurs prises en charge à la liste, vous devez modifier manuellement le fichier \Data\Std\stdConnProps.xml dans le répertoire produit (<ProductDir>). Pour plus d'informations, voir l'annexe Connector Configurator de ce guide.

CommonEventInfrastructure

L'infrastructure Common Event Infrastructure (CEI) est une fonction simple de gestion des événements chargée de traiter les événements générés. La propriété CommonEventInfrastructure indique si le CEI doit être appelé lors de l'exécution.

La valeur par défaut est false.

CommonEventInfrastructureContextURL

CommonEventInfrastructureContextURL est utilisé pour accéder au serveur WAS qui exécute l'application du serveur CEI (Common Event Infrastructure). Cette propriété précise l'URL à utiliser.

Cette propriété n'est valide que si la valeur de CommonEventInfrastructure est définie sur true.

La valeur par défaut est une zone vide.

ConcurrentEventTriggeredFlows

La propriété ConcurrentEventTriggeredFlows détermine le nombre d'objets métier pouvant être traités simultanément par le connecteur pour la transmission des événements. Vous définissez la valeur de cet attribut sur le nombre d'objets métiers qui sont simultanément mappés et livrés. Par exemple, si vous définissez la valeur de cette propriété sur 5, cinq objets métier sont traités simultanément.

La définition de cette propriété sur une valeur supérieure à 1 permet au connecteur d'une application source de mapper plusieurs objets métier d'événement en même temps et de les transmettre simultanément à plusieurs instances de collaboration. Cela augmente la rapidité de transmission des objets métier au courtier d'intégration, en particulier si les objets métier utilisent des mappes complexes. L'augmentation du taux d'arrivée des objets métier aux instances de collaboration peut améliorer les performances générales du système.

Pour implémenter le traitement simultané d'un flux entier (d'une application source vers une application cible), vous devez configurer les propriétés suivantes :

- La collaboration doit être configurée de façon à utiliser plusieurs unités d'exécution simultanées, en indiquant pour la propriété Maximum number of concurrent events une valeur suffisamment élevée.
- Le composant spécifique à l'application de destination doit être configuré pour traiter les requêtes simultanément. C'est à dire qu'il doit avoir plusieurs unités d'exécution ou être capable d'utiliser le parallélisme de l'agent du connecteur et être configuré pour plusieurs processus. Attribuez une valeur supérieure à 1 à la propriété de configuration Parallel Process Degree.

La propriété ConcurrentEventTriggeredFlows property n'a aucun effet sur l'interrogation du connecteur, laquelle n'a qu'une seule unité d'exécution et est exécutée en série.

La propriété n'est valide que si la valeur de la propriété RepositoryDirectory est définie sur <REMOTE>.

La valeur par défaut est 1.

ContainerManagedEvents

La propriété ContainerManagedEvents permet à un connecteur activé par JMS et utilisant un magasin d'événements JMS d'effectuer une transmission garantie d'événement, dans laquelle un événement est retiré de la file d'attente source et placé sur la file d'attente cible en tant qu'une transaction JMS.

Lorsque cette propriété est définie sur JMS, les propriétés suivantes doivent également être définies pour activer la transmission garantie d'événement :

- PollQuantity = 1 à 500
- SourceQueue = /SOURCEQUEUE

Vous devez aussi configurer un gestionnaire de données avec les propriétés MimeType et DHClass (classe de gestionnaire de données). Vous pouvez également ajouter DataHandlerConfigMOName (le nom de méta-objet facultatif). Pour définir ces valeurs, utilisez l'onglet **Data Handler** dans Connector Configurator.

Bien que ces propriétés soient spécifiques à l'adaptateur, voici quelques exemples de valeurs :

- MimeType = text/xml
- DHClass = com.crossworlds.DataHandlers.text.xml
- DataHandlerConfigMOName = MO_DataHandler_Default

Les zones qui correspondent à ces valeurs dans l'onglet **Data Handler** ne s'affichent que si vous avez défini la propriété ContainerManagedEvents sur la valeur JMS.

Remarque : Lorsque ContainerManagedEvents a la valeur JMS, le connecteur n'appelle pas sa méthode pollForEvents(), ce qui en désactive la fonctionnalité.

La propriété ContainerManagedEvents n'est valide que si la valeur de la propriété DeliveryTransport est définie sur JMS.

Il n'y a pas de valeur par défaut.

ControllerEventSequencing

La propriété ControllerEventSequencing autorise le séquençage des événements dans le contrôleur du connecteur.

Cette propriété n'est valide que si la valeur de la propriété RepositoryDirectory est définie sur <REMOTE> (BrokerType égal à ICS).

La valeur par défaut est true.

ControllerStoreAndForwardMode

La propriété ControllerStoreAndForwardMode définit le comportement du contrôleur du connecteur après avoir détecté l'indisponibilité du composant spécifique à l'application cible.

Si cette propriété a la valeur true et que le composant spécifique à l'application cible n'est pas disponible lorsqu'un événement atteint l'ICS, le contrôleur du connecteur empêche la requête d'accéder au composant spécifique à l'application. Lorsque le composant spécifique à l'application redevient opérationnel, le contrôleur lui envoie la requête.

Toutefois, si le composant d'application cible devient indisponible après que le contrôleur du connecteur lui a envoyé la requête d'appel de service, celle-ci échoue.

Si cette propriété a la valeur `false`, le contrôleur du connecteur met toutes les requêtes d'appels de service en échec dès qu'il détecte l'indisponibilité du composant spécifique à l'application.

Cette propriété n'est valide que si la valeur de la propriété `RepositoryDirectory` est définie sur `<REMOTE>` (la valeur de la propriété `BrokerType` est `ICS`).

La valeur par défaut est `true`.

ControllerTraceLevel

La propriété `ControllerTraceLevel` définit le niveau des messages de trace pour le contrôleur de connecteur.

La propriété n'est valide que si la valeur de la propriété `RepositoryDirectory` est définie sur `<REMOTE>`.

La valeur par défaut est `0`.

DeliveryQueue

La propriété `DeliveryQueue` définit la file d'attente utilisée par le connecteur pour envoyer des objets métier au courtier d'intégration.

Cette propriété n'est valide que si la valeur de la propriété `DeliveryTransport` est définie sur `JMS`.

La valeur par défaut est `<CONNECTORNAME>/DELIVERYQUEUE`.

DeliveryTransport

La propriété `DeliveryTransport` spécifie le mécanisme de transfert pour la transmission des événements. Les valeurs possibles sont `MQ` pour `WebSphere MQ`, `IDL` pour `CORBA IIOP` et `JMS` pour `Java Messaging Service`.

- Si la valeur de la propriété `RepositoryDirectory` est définie sur `<REMOTE>`, la valeur de la propriété `DeliveryTransport` peut être `MQ`, `IDL` ou `JMS`, et la valeur par défaut est `IDL`.
- Si la valeur de la propriété `RepositoryDirectory` est un répertoire local, l'unique valeur possible est `JMS`.

Si la valeur de la propriété `RepositoryDirectory` est `MQ` ou `IDL`, le connecteur envoie des requêtes d'appel de service et des messages administratifs par `CORBA IIOP`.

Si la valeur de la propriété `DeliveryTransport` est `MQ`, vous pouvez définir le paramètre de ligne de commande `WhenServerAbsent` dans le script de démarrage de l'adaptateur, de façon à indiquer si l'adaptateur doit se mettre en pause ou se fermer lorsque `InterChange Server` est arrêté.

- Entrez `WhenServerAbsent=pause` pour mettre l'adaptateur en pause lorsque `ICS` n'est pas disponible.
- Entrez `WhenServerAbsent=shutdown` pour arrêter l'adaptateur lorsque `ICS` n'est pas disponible.

WebSphere MQ et IDL

Utilisez WebSphere MQ plutôt que IDL pour le transfert d'événement, sauf si vous ne devez avoir qu'un seul produit. WebSphere MQ présente les avantages suivants par rapport à IDL :

- Communication asynchrone :
WebSphere MQ permet au composant spécifique à l'application d'interroger et de stocker de manière permanente les événements, même lorsque le serveur n'est pas disponible.
- Performance côté serveur :
WebSphere MQ offre plus de rapidité du côté du serveur. En mode optimisé, WebSphere MQ ne stocke que le pointeur désignant un événement dans la base de données du référentiel, tandis que l'événement correspondant reste dans la file d'attente de WebSphere MQ. Ceci empêche d'écrire des événements potentiellement volumineux dans la base de données du référentiel.
- Performance côté agent :
WebSphere MQ offre plus de rapidité du côté du composant spécifique à l'application. Avec WebSphere MQ, l'unité d'exécution d'interrogation du connecteur sélectionne un événement, le place dans la file d'attente du connecteur, puis sélectionne l'événement suivant. Cette méthode est plus rapide que celle d'IDL, dans laquelle l'unité d'exécution d'interrogation du connecteur doit sélectionner un événement, accéder au réseau dans le processus du serveur, stocker l'événement de manière permanente dans la base de données du référentiel, puis sélectionner l'événement suivant.

JMS

Le mécanisme de transfert JMS active la communication entre le connecteur et l'architecture du connecteur client à l'aide de Java Messaging Service (JMS).

Si vous sélectionnez JMS en tant que transfert, d'autres propriétés JMS supplémentaires telles que `jms.MessageBrokerName`, `jms.FactoryClassName`, `jms.Password` et `jms.UserName` apparaissent dans Connector Configurator. Les propriétés `jms.MessageBrokerName` et `jms.FactoryClassName` sont obligatoires pour ce transfert.

Il peut y avoir une limitation de mémoire si vous utilisez le mécanisme de transfert JMS pour un connecteur dans l'environnement suivant :

- AIX 5.0
- WebSphere MQ 5.3.0.1
- ICS est le courtier d'intégration

Dans cet environnement, vous rencontrerez peut être des difficultés pour démarrer simultanément le contrôleur du connecteur (du côté du serveur) et le connecteur (du côté du client), en raison de l'utilisation de la mémoire dans le client WebSphere MQ. Si votre installation utilise une taille de segment de processus inférieure à 768 Mo, définissez la variable et la propriété suivantes :

- Définissez la variable d'environnement `LDR_CNTRL` dans le script `CWSharedEnv.sh`.

Ce script se trouve dans le répertoire `\bin` sous le répertoire produit (`<ProductDir>`). A l'aide d'un éditeur de texte, ajoutez la ligne suivante à la première ligne du script `CWSharedEnv.sh` :

```
export LDR_CNTRL=MAXDATA=0x30000000
```

Cette ligne de commande restreint l'utilisation du segment de mémoire à un maximum de 768 Mo (3 segments * 256 Mo). Si la mémoire du processus dépasse cette limite, un échange de pages peut se produire, ce qui peut affecter les performances de votre système.

- Définissez la valeur de la propriété `IPCCBaseAddress` sur 11 ou 12. Pour plus d'informations sur cette propriété, voir le document *System Installation Guide for UNIX*.

DuplicateEventElimination

Lorsque la valeur de cette propriété est `true`, un connecteur activé pour JMS peut vérifier que des doublons ne sont pas transmis à la file d'attente de transmission. Pour utiliser cette fonction, le connecteur doit recevoir pendant son développement un identificateur d'événement unique défini en tant qu'attribut `ObjectEventId` de l'objet métier dans le code spécifique à l'application.

Remarque : Lorsque la valeur de cette propriété est `true`, la propriété `MonitorQueue` doit être activée pour garantir la livraison de l'événement.

La valeur par défaut est `false`.

EnableOidForFlowMonitoring

Lorsque la valeur de cette propriété est `true`, l'exécution de l'adaptateur marque le `ObjectEventID` entrant en tant que clé étrangère pour la surveillance du flot.

La propriété n'est valide que si la propriété `BrokerType` est définie sur `ICS`.

La valeur par défaut est `false`.

FaultQueue

Si le connecteur rencontre une erreur lors du traitement d'un message, il transmet ce message à la file d'attente indiquée dans la propriété `FaultQueue` (accompagné d'un indicateur de statut et d'une description de l'incident).

La valeur par défaut est `<CONNECTORNAME>/FAULTQUEUE`.

jms.FactoryClassName

La propriété `jms.FactoryClassName` indique le nom de classe à instancier pour un fournisseur JMS. Cette propriété doit être définie si la valeur de la propriété `DeliveryTransportProperty` est `JMS`.

La valeur par défaut est `CxCommon.Messaging.jms.IBMMQSeriesFactory`.

jms.ListenerConcurrency

La propriété `jms.ListenerConcurrency` indique le nombre de programmes d'écoute simultanés pour le contrôleur JMS. Elle précise le nombre d'unités d'exécution qui extraient et traitent les messages simultanément, dans un contrôleur.

Cette propriété n'est valide que si la valeur de la propriété `jms.OptimizedTransport` est `true`.

La valeur par défaut est `1`.

jms.MessageBrokerName

`jms.MessageBrokerName` précise le nom de courtier à utiliser pour le fournisseur JMS. Vous devez définir cette propriété de connecteur si vous précisez JMS en tant que mécanisme de transfert (dans la propriété `DeliveryTransport`).

Lorsque vous vous connectez à un courtier de messages éloigné, cette propriété exige les valeurs suivantes :

QueueMgrName:Channel:HostName:PortNumber

où :

QueueMgrName est le nom du gestionnaire de files d'attente.

Channel est le canal utilisé par le client.

HostName est le nom de la machine sur laquelle doit résider le gestionnaire de files d'attente.

PortNumber est le numéro de port sur lequel écoute le gestionnaire de files d'attente.

Par exemple :

```
jms.MessageBrokerName = WBIMB.Queue.Manager:CHANNEL1:RemoteMachine:1456
```

La valeur par défaut est `crossworlds.queue.manager`. Utilisez la valeur par défaut lorsque vous vous connectez à un courtier de messages local.

jms.NumConcurrentRequests

La propriété `jms.NumConcurrentRequests` indique le nombre maximal de requêtes d'appel de service pouvant être envoyées simultanément à un connecteur. Lorsque ce nombre maximal est atteint, les nouveaux appels de service sont bloqués et mis en attente de traitement.

La valeur par défaut est 10.

jms.Password

La propriété `jms.Password` indique le mot de passe défini pour le fournisseur JMS. Cette valeur est facultative.

Il n'y a pas de valeur par défaut.

jms.TransportOptimized

La propriété `jms.TransportOptimized` détermine si l'opération en cours (WIP) est optimisée. Pour l'optimiser, vous devez disposer d'un fournisseur WebSphere MQ. Pour que le WIP optimisé fonctionne, le fournisseur de messagerie doit pouvoir :

1. Lire un message sans le retirer de la file d'attente
2. Supprimer un message ayant un ID donné sans transférer le message entier dans l'espace mémoire du réceptionnaire
3. Lire un message en utilisant un ID donné (nécessaire pour la récupération)
4. Déterminer à quel moment apparaissent les événements qui n'ont pas été lus.

Les API JMS ne peuvent pas être utilisées pour le WIP optimisé car elles ne remplissent pas les conditions 2 et 4 ci-dessus. Les API MQ Java remplissent les quatre conditions et sont donc requises pour le WIP optimisé.

Cette propriété n'est valide que si la valeur de `DeliveryTransport` est JMS et la valeur de `BrokerType` est ICS.

La valeur par défaut est false.

jms.UserName

La propriété `jms.UserName` indique le nom d'utilisateur du fournisseur JMS. Cette valeur est facultative.

Il n'y a pas de valeur par défaut.

JvmMaxHeapSize

La propriété `JvmMaxHeapSize` indique la taille de segment maximale pour l'agent (en megaoctets).

La propriété n'est valide que si la valeur de la propriété `RepositoryDirectory` est définie sur `<REMOTE>`.

La valeur par défaut est 128 Mo.

JvmMaxNativeStackSize

La propriété `JvmMaxNativeStackSize` indique l'espace mémoire natif maximal pour l'agent (en kilo-octets).

La propriété n'est valide que si la valeur de la propriété `RepositoryDirectory` est définie sur `<REMOTE>`.

La valeur par défaut est 128 Ko.

JvmMinHeapSize

La propriété `JvmMinHeapSize` indique la taille de segment minimale pour l'agent (en megaoctets).

La propriété n'est valide que si la valeur de la propriété `RepositoryDirectory` est définie sur `<REMOTE>`.

La valeur par défaut est 1 Mo.

ListenerConcurrency

La propriété `ListenerConcurrency` prend en charge le traitement de plusieurs unités d'exécution dans WebSphere MQ Listener lorsque ICS est le courtier d'intégration. Elle permet l'écriture par lots de plusieurs événements sur la base de données, ce qui améliore les performances du système.

Cette propriété n'est valide que pour les connecteurs qui utilisent le transfert MQ. La valeur de la propriété `DeliveryTransport` doit être définie sur MQ.

La valeur par défaut est 1.

Locale

La propriété `Locale` indique le code de langue, le pays ou le territoire et, le cas échéant, le jeu de codes de caractères associé. La valeur de cette propriété détermine les conventions culturelles telles que le classement et l'ordre de tri des données, les formats de date et d'heure, ainsi que les symboles monétaires utilisés.

Le format d'un nom d'environnement local est le suivant :

ll_TT.codeset

où :

ll est un code de langue à deux caractères (en minuscules)

TT est un code pays ou territoire à deux caractères (en majuscules)

codeset est le nom du jeu de codes de caractères associé (facultatif).

Par défaut, n'est affiché qu'un sous-ensemble des paramètres régionaux pris en charge. Pour ajouter d'autres valeurs prises en charge à la liste, modifiez le fichier `\Data\Std\stdConnProps.xml` dans le répertoire `<ProductDir>\bin`. Pour plus d'informations, voir l'annexe Connector Configurator de ce guide.

Si le connecteur n'a pas été internationalisé, la seule valeur correcte pour cette propriété est `en_US`. Pour déterminer si un connecteur spécifique a été internationalisé, consultez le guide utilisateur de l'adaptateur.

La valeur par défaut est `en_US`.

LogAtInterchangeEnd

La propriété `LogAtInterchangeEnd` indique s'il faut consigner les erreurs dans le journal du courtier d'intégration.

La consignation des erreurs dans le journal active également la notification par courrier électronique qui, lorsque des erreurs ou erreurs fatales ont lieu, génère des messages électroniques pour le destinataire spécifié par la valeur `MESSAGE_RECIPIENT` dans le fichier `InterchangeSystem.cfg`. Par exemple, lorsque la connexion entre un connecteur et son application est interrompue, si la valeur de `LogAtInterChangeEnd` est `true`, un courrier électronique est envoyé au destinataire indiqué.

Cette propriété n'est valide que si la valeur de la propriété `RespositoryDirectory` est définie sur `<REMOTE>` (la valeur de `BrokerType` est `ICS`).

La valeur par défaut est `false`.

MaxEventCapacity

La propriété `MaxEventCapacity` indique le nombre maximal d'événements contenus dans la mémoire tampon du contrôleur. Cette propriété est utilisée par la fonctionnalité de contrôle de flux.

Cette propriété n'est valide que si la valeur de la propriété `RespositoryDirectory` est définie sur `<REMOTE>` (la valeur de `BrokerType` est `ICS`).

La valeur peut être un nombre entier positif compris entre 1 et 2147483647.

La valeur par défaut est 2147483647.

MessageFileName

La propriété `MessageFileName` indique le nom du fichier de messages du connecteur. L'emplacement standard de ce fichier de messages est `\connectors\messages`, dans le répertoire produit. Si le fichier de messages n'est pas situé à l'emplacement standard, indiquez son nom dans un chemin d'accès absolu.

S'il n'existe pas de fichier de messages, le connecteur utilise `InterchangeSystem.txt` comme fichier de messages. Ce fichier est situé dans le répertoire produit.

Remarque : Pour déterminer si un connecteur a son propre fichier de messages, reportez-vous au guide d'utilisation de l'adaptateur.

La valeur par défaut est `InterchangeSystem.txt`.

MonitorQueue

La propriété `MonitorQueue` indique la file d'attente logique utilisée par le connecteur pour surveiller les événements en double.

Elle n'est valide que si la valeur de la propriété `DeliveryTransport` est `JMS` et la valeur de `DuplicateEventElimination` est `true`.

La valeur par défaut est `<CONNECTORNAME>/MONITORQUEUE`

OADAutoRestartAgent

La propriété `OADAutoRestartAgent` indique si le connecteur utilise la fonction de redémarrage automatique et éloigné. Cette fonction utilise le démon d'activation d'objets (OAD, Object Activation Daemon) déclenché par WebSphere MQ pour redémarrer le connecteur après un arrêt anormal ou pour démarrer un connecteur éloigné à partir du moniteur système.

Cette propriété doit avoir la valeur `true` pour que la fonction de redémarrage automatique et à distance soit activée. Pour plus d'informations sur la configuration de la fonction de l'OAD déclenché par WebSphere MQ, voir le document *Installation Guide for Windows* ou *for UNIX*.

Cette propriété n'est valide que si la valeur de la propriété `RespositoryDirectory` est définie sur `<REMOTE>` (la valeur de `BrokerType` est `ICS`).

La valeur par défaut est `false`.

OADMaxNumRetry

La propriété `OADMaxNumRetry` indique le nombre maximal de tentatives de redémarrage du connecteur après un arrêt anormal, automatiquement tentées par l'OAD déclenché par WebSphere MQ. Pour cela, la propriété `OADAutoRestartAgent` doit avoir la valeur `true`.

Cette propriété n'est valide que si la valeur de la propriété `RespositoryDirectory` est définie sur `<REMOTE>` (la valeur de `BrokerType` est `ICS`).

La valeur par défaut est `1000`.

OADRetryTimeInterval

La propriété `OADRetryTimeInterval` indique pour l'OAD déclenché par WebSphere MQ la durée en minutes entre les tentatives de relance. Si l'agent du connecteur ne redémarre pas durant cet intervalle, le contrôleur du connecteur demande à l'OAD de redémarrer l'agent du connecteur. L'OAD répète cette opération autant de fois que spécifié par la propriété `OADMaxNumRetry`. Pour cela, la propriété `OADAutoRestartAgent` doit avoir la valeur `true`.

Cette propriété n'est valide que si la valeur de la propriété RespositoryDirectory est définie sur <REMOTE> (la valeur de BrokerType est ICS).

La valeur par défaut est 10.

PollEndTime

La propriété PollEndTime indique l'heure d'arrêt de l'interrogation de la file d'attente des événements. Le format est *HH:MM*, dans lequel *HH* représente les heures (de 0 à 23) et *MM* représente les secondes (de 0 à 59).

Vous devez saisir une valeur correcte pour cette propriété. La valeur par défaut est HH:MM sans valeur indiquée, mais elle doit être précisée.

Si l'exécution de l'adaptateur détecte :

- que PollStartTime est défini et PollEndTime n'est pas défini, ou
- que PollEndTime est défini et PollStartTime n'est pas défini,

l'interrogation utilisera la valeur configurée pour la propriété PollFrequency.

PollFrequency

La propriété PollFrequency indique la durée (en millisecondes) entre la fin de la dernière interrogation et le début de la suivante. Il ne s'agit pas de l'intervalle entre chaque interrogation. Le principe est le suivant :

- Lancez une interrogation pour obtenir le nombre d'objets spécifié par la propriété PollQuantity.
- Traitez ces objets. Pour certains connecteurs, ceci peut se faire en partie sur des unités d'exécution séparées, qui s'exécutent de manière asynchrone jusqu'à l'interrogation suivante.
- Attendez pendant l'intervalle indiqué par la propriété PollFrequency.
- Répétez le cycle.

La valeurs suivantes sont valides pour cette propriété :

- Un nombre de millisecondes (un entier positif).
- Le mot *no*, pour que le connecteur n'émette pas d'interrogation. Saisissez le mot en minuscules.
- Le mot *key*, pour que le connecteur émette des interrogations uniquement lorsque vous tapez la lettre *p* dans la fenêtre d'invite de commande du connecteur. Saisissez le mot en minuscules.

La valeur par défaut est 10000.

Important : Certains connecteurs sont limités dans l'utilisation de cette propriété. Ces restrictions sont décrites dans le chapitre sur l'installation et la configuration de l'adaptateur.

PollQuantity

La propriété PollQuantity désigne le nombre d'éléments de l'application pour lesquels le connecteur doit émettre des interrogations. Si l'adaptateur dispose d'une propriété spécifique au connecteur pour définir le nombre d'interrogations, la valeur définie dans cette propriété spécifique remplace la valeur de la propriété standard.

Cette propriété n'est valide que si la valeur de la propriété `DeliveryTransport` est `JMS`, et si la propriété `ContainerManagedEvents` a une valeur.

Un message électronique est également considéré comme étant un événement. Lorsqu'il est interrogé pour un courrier électronique, le connecteur agit comme suit :

- Lorsqu'il est interrogé une fois, le connecteur détecte le corps du message, qu'il lit comme une pièce jointe. Comme aucun gestionnaire de données n'a été spécifié pour ce type mime, il ignorera le message.
- Le connecteur traite la première pièce jointe BO. Le gestionnaire de données est disponible pour ce type MIME : l'objet métier est envoyé à Visual Test Connector.
- Lorsqu'il est interrogé pour la deuxième fois, le connecteur traite la deuxième pièce jointe BO. Le gestionnaire de données est disponible pour ce type MIME : l'objet métier est envoyé à Visual Test Connector.
- Une fois acceptée, la troisième pièce jointe BO peut être transmise.

PollStartTime

La propriété `PollStartTime` indique l'heure de démarrage de l'interrogation de la file d'attente des événements. Le format est `HH:MM`, dans lequel `HH` représente les heures (de 0 à 23) et `MM` représente les secondes (de 0 à 59).

Vous devez saisir une valeur correcte pour cette propriété. La valeur par défaut est `HH:MM` sans valeur indiquée, mais elle doit être modifiée.

Si l'exécution de l'adaptateur détecte :

- que `PollStartTime` est défini et `PollEndTime` n'est pas défini, ou
- que `PollEndTime` est défini et `PollStartTime` n'est pas défini,

l'interrogation utilisera la valeur configurée pour la propriété `PollFrequency`.

RepositoryDirectory

La propriété `RepositoryDirectory` indique l'emplacement du référentiel à partir duquel le connecteur lit les schémas XML qui stockent les métadonnées pour la définition des objets métier.

Si ICS est le courtier d'intégration, cette valeur doit être définie sur `<REMOTE>`, car le connecteur obtient ces informations à partir du référentiel d'InterChange Server.

Lorsque le courtier d'intégration est un courtier de message WebSphere ou WAS, cette valeur est définie par défaut sur `<ProductDir>\repository`. Toutefois, elle peut être définie sur tout nom valide de répertoire.

RequestQueue

La propriété `RequestQueue` précise la file d'attente utilisée par le courtier d'intégration pour envoyer des objets métier au connecteur.

Cette propriété n'est valide que si la valeur de la propriété `DeliveryTransport` est `JMS`.

La valeur par défaut est `<CONNECTORNAME>/REQUESTQUEUE`.

ResponseQueue

La propriété ResponseQueue désigne la file d'attente de réponses JMS, qui transmet un message de réponse depuis l'architecture du connecteur vers le courtier d'intégration. Lorsqu'ICS est le courtier d'intégration, le serveur envoie la requête et attend un message de réponse dans la file d'attente de réponses JMS.

Cette propriété n'est valide que si la valeur de la propriété DeliveryTransport est définie sur JMS.

La valeur par défaut est <CONNECTORNAME>/RESPONSEQUEUE.

RestartRetryCount

La propriété RestartRetryCount indique le nombre de tentatives de redémarrage du connecteur. Lorsqu'elle est utilisée pour un connecteur parallèle, cette propriété indique le nombre de tentatives de redémarrage du composant spécifique à l'application du connecteur client par le composant spécifique à l'application du connecteur maître.

La valeur par défaut est 7.

RestartRetryInterval

La propriété RestartRetryInterval indique l'intervalle en minutes pendant lequel le connecteur tente de se redémarrer. Lorsqu'elle est utilisée pour un connecteur parallèle, cette propriété indique l'intervalle entre les tentatives de redémarrage du composant spécifique à l'application du connecteur client par le composant spécifique à l'application du connecteur maître.

Les valeurs possibles vont de 1 à 2147483647.

La valeur par défaut est 1.

ResultSetEnabled

La propriété ResultSetEnabled active ou désactive la prise en charge de l'ensemble des résultats lorsque Information Integrator est actif. Cette propriété ne peut être utilisée que si l'adaptateur prend en charge DB2 Information Integrator.

Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS et la valeur de BrokerType est WMQI.

La valeur par défaut est false.

ResultSetSize

La propriété ResultSetSize définit le nombre maximum d'objets métier pouvant être retournés à Information Integrator. Cette propriété ne peut être utilisée que si l'adaptateur prend en charge DB2 Information Integrator.

Cette propriété n'est valide que si la propriété ResultSetEnabled est définie sur true.

La valeur par défaut est 0. Ce qui signifie que la taille de l'ensemble de résultats est illimitée.

RHF2MessageDomain

La propriété RHF2MessageDomain vous permet de configurer la valeur du nom de domaine de la zone dans l'en-tête JMS. Lorsque les données sont envoyées à un courtier de message WebSphere par transfert JMS, l'architecture de l'adaptateur écrit les informations de l'en-tête JMS, avec un nom de domaine et une valeur fixe mrm. Un nom de domaine configurable vous permet d'analyser comment le courtier de messages WebSphere traite les données de message.

Voici un exemple d'en-tête :

```
<mcd><Msd>mrm</Msd><Set>3</Set><Type>
Retek_POPhyDesc</Type><Fmt>CwXML</Fmt></mcd>
```

Cette propriété n'est valide que si la valeur de BrokerType est WMQI ou WAS. De plus, elle n'est valide que si la valeur de la propriété DeliveryTransport est JMS et si la valeur de WireFormat est CwXML.

Les valeurs possibles sont mrm et xml. La valeur par défaut est mrm.

SourceQueue

La propriété SourceQueue désigne la file d'attente source JMS de l'architecture du connecteur qui assure la transmission garantie d'événements pour les connecteur activés par JMS qui utilisent un magasin d'événements JMS. Pour plus d'informations, voir «ContainerManagedEvents», à la page 58.

Cette propriété n'est valide que si la valeur de DeliveryTransport est JMS et si une valeur est indiquée pour ContainerManagedEvents.

La valeur par défaut est <CONNECTORNAME>/SOURCEQUEUE.

SynchronousRequestQueue

La propriété SynchronousRequestQueue transmet les messages de requête qui requièrent une réponse synchrone depuis l'architecture du connecteur vers le courtier. Cette file d'attente n'est nécessaire que si le connecteur utilise l'exécution synchrone. Avec l'exécution synchrone, l'architecture du connecteur envoie un message à la file d'attente de requêtes synchrones et attend une réponse du courtier sur la file d'attente de réponse synchrone. La réponse envoyée au connecteur a un ID de corrélation qui correspond à l'ID du message d'origine.

Cette propriété n'est valide que si la valeur de la propriété DeliveryTransport est définie sur JMS.

La valeur par défaut est <CONNECTORNAME>/SYNCHRONOUSREQUESTQUEUE

SynchronousRequestTimeout

La propriété SynchronousRequestTimeout indique la durée d'attente en millisecondes d'une réponse à une requête synchrone. Si la réponse n'est pas reçue dans l'intervalle de temps indiqué, le connecteur transfère le message de requête synchrone original, accompagné d'un message d'erreur, dans la file d'attente des erreurs.

Cette propriété n'est valide que si la valeur de la propriété DeliveryTransport est définie sur JMS.

La valeur par défaut est 0.

SynchronousResponseQueue

La propriété SynchronousResponseQueue transmet les messages de réponse à une requête synchrone entre le courtier et l'architecture du connecteur. Cette file d'attente n'est nécessaire que si le connecteur utilise l'exécution synchrone.

Cette propriété n'est valide que si la valeur de la propriété DeliveryTransport est définie sur JMS.

La valeur par défaut est <CONNECTORNAME>/SYNCHRONOUSRESPONSEQUEUE

TivoliMonitorTransactionPerformance

La propriété TivoliMonitorTransactionPerformance indique si IBM Tivoli Monitoring for Transaction Performance (ITMTP) est appelé lors de l'exécution.

La valeur par défaut est false.

WireFormat

La propriété WireFormat précise le format de message sur le transfert :

- Si la valeur de la propriété RepositoryDirectory est un répertoire local, la valeur est CwXML.
- Si la valeur de la propriété RepositoryDirectory est un répertoire éloigné, la valeur est CwB0.

WsifSynchronousRequestTimeout

La propriété WsifSynchronousRequestTimeout indique la durée d'attente en millisecondes d'une réponse à une requête synchrone. Si la réponse n'est pas reçue dans l'intervalle de temps indiqué, le connecteur transfère le message de requête synchrone original, accompagné d'un message d'erreur, dans la file d'attente des erreurs.

Cette propriété n'est valide que si la valeur de BrokerType est définie sur WAS.

La valeur par défaut est 0.

XMLNameSpaceFormat

La propriété XMLNameSpaceFormat précise des espaces de nom courts ou longs dans le format XML des définitions d'objet métier.

Cette propriété n'est valide que si la valeur de BrokerType est WMQI ou WAS.

La valeur par défaut est short.

Annexe B. Connector Configurator

Cette annexe décrit comment utiliser Connector Configurator afin de définir les valeurs des propriétés de configuration pour votre adaptateur.

Connector Configurator vous permet de :

- créer un modèle de propriété spécifique au connecteur pour la configuration de votre connecteur ;
- créer un fichier de configuration ;
- définir les propriétés dans un fichier de configuration.

Les sujets traités dans cette annexe sont les suivants :

- «Présentation de Connector Configurator», à la page 73
- «Démarrage de Connector Configurator», à la page 74
- «Création d'un modèle de propriétés spécifiques au connecteur», à la page 75
- «Création d'un fichier de configuration», à la page 78
- «Définition des propriétés d'un fichier de configuration», à la page 81
- «Utilisation de Connector Configurator dans un environnement globalisé», à la page 91

Présentation de Connector Configurator

Connector Configurator vous permet de configurer le connecteur de votre adaptateur à utiliser avec ces courtiers d'intégration :

- WebSphere InterChange Server (ICS) ;
- WebSphere MQ Integrator, WebSphere MQ Integrator Broker et WebSphere Business Integration Message Broker, appelés courtiers de messages WebSphere (WMQI) ;
- WebSphere Application Server (WAS).

Si votre adaptateur prend en charge DB2 Information Integrator, utilisez les options et les propriétés standard DB2 II (voir la colonne Remarque de l'annexe Propriétés standard).

Connector Configurator vous permet de :

- créer un **modèle de propriété spécifique au connecteur** pour la configuration de votre connecteur ;
- créer un **fichier de configuration du connecteur** (vous devez créer un fichier de configuration pour chaque connecteur installé) ;
- définir les propriétés dans un fichier de configuration.
Vous devez peut-être modifier les valeurs par défaut définies pour les propriétés dans les modèles du connecteur. Vous devez également déterminer les définitions d'objet métier prises en charge, indiquer les mappes à utiliser avec les collaborations à l'aide d'ICS et spécifier les paramètres d'application de messagerie, de journalisation, de trace ainsi que ceux du gestionnaire de données, le cas échéant.

Le mode dans lequel vous exécutez Connector Configurator et le type de fichier de configuration que vous utilisez peuvent différer en fonction du courtier

d'intégration que vous exécutez. Par exemple, si vous utilisez WMQI comme courtier, vous exécutez Connector Configurator directement, et non à partir de System Manager (voir «Exécution de Connector Configurator en mode autonome», à la page 74).

Les propriétés de configuration du connecteur incluent des propriétés de configuration standard (les propriétés communes à tous les connecteurs) et des propriétés spécifiques au connecteur (propriétés requises par le connecteur pour une technologie ou une application spécifique).

Dans la mesure où les **propriétés standard** sont utilisées par tous les connecteurs, vous n'avez pas besoin de définir ces propriétés de tout pièce ; Connector Configurator les incorpore à votre fichier de configuration dès que vous créez ce fichier. Cependant, vous devez définir la valeur de chaque propriété standard dans Connector Configurator.

L'intervalle des propriétés standard peut être différent pour tous les courtiers et toutes les configurations. Certaines propriétés ne sont disponibles que si vous attribuez une valeur spécifique à d'autres propriétés. La fenêtre des propriétés standard dans Connector Configurator affiche les propriétés disponibles pour votre configuration spécifique.

Cependant, pour les **propriétés spécifiques au connecteur**, vous devez d'abord définir les propriétés, puis leur attribuer une valeur. Pour ce faire, créez un modèle de propriétés spécifiques au connecteur pour votre adaptateur particulier. Il se peut qu'un modèle soit déjà configuré dans votre système, auquel cas vous pouvez l'utiliser. Dans le cas contraire, suivez les étapes dans la section «Création d'un modèle», à la page 76 pour configurer un nouveau modèle.

Utilisation des connecteurs sous UNIX

Connector Configurator s'exécute uniquement dans un environnement Windows. Si vous exécutez le connecteur dans un environnement UNIX, utilisez Connector Configurator dans Windows pour modifier le fichier de configuration, puis copiez le fichier dans votre environnement UNIX.

Certaines propriétés de Connector Configurator utilisent des chemins de répertoire, par défaut avec la convention de dénomination propre à Windows. Si vous utilisez le fichier de configuration sous UNIX, vous devrez modifier les chemins d'accès aux répertoires pour qu'ils respectent les conventions UNIX. Afin d'activer les bonnes règles de système d'exploitation pour la validation étendue, sélectionnez le système d'exploitation cible dans la liste déroulante de la barre d'outils.

Démarrage de Connector Configurator

Vous pouvez démarrer et exécuter Connector Configurator dans l'un de ces deux modes :

- de manière indépendante, en mode autonome ;
- à partir de System Manager.

Exécution de Connector Configurator en mode autonome

Vous pouvez exécuter Connector Configurator en mode autonome, sans exécuter le System Manager, et utiliser les fichiers de configuration quel que soit votre courtier.

Pour ce faire, procédez comme suit :

- Dans **Démarrer>Programmes**, cliquez sur **IBM WebSphere Business Integration Adapters>IBM WebSphere Business Integration Toolset>Connector Configurator**.
- Sélectionnez **File>New>Connector Configuration**.
- Lorsque vous cliquez sur le menu déroulant en regard de **System Connectivity Integration Broker**, vous pouvez sélectionner ICS, WebSphere Message Brokers ou WAS, en fonction de votre courtier.

Vous pouvez choisir d'exécuter Connector Configurator en mode autonome pour créer le fichier, puis de vous connecter à System Manager afin de l'enregistrer dans un projet System Manager (voir «Remplissage d'un fichier de configuration», à la page 81).

Exécution de Connector Configurator à partir de System Manager

Vous pouvez exécuter Connector Configurator à partir de System Manager.

Pour exécuter Connector Configurator, procédez comme suit :

1. Ouvrez System Manager.
2. Dans la fenêtre System Manager, développez l'icône **Integration Component Libraries** et mettez en évidence **Connecteurs**.
3. Dans la barre de menus de System Manager, cliquez sur **Outils>Connector Configurator**. La fenêtre Connector Configurator affiche la boîte de dialogue **New Connector**.
4. Lorsque vous cliquez sur le menu déroulant en regard de **System Connectivity Integration Broker**, vous pouvez sélectionner ICS, WebSphere Message Brokers ou WAS, en fonction de votre courtier.

Pour modifier un fichier de configuration existant, procédez comme suit :

- Dans la fenêtre System Manager, sélectionnez l'un des fichiers de configuration répertoriés dans le dossier du connecteur et cliquez dessus avec le bouton droit. Connector Configurator affiche le fichier de configuration avec le type de courtier d'intégration et le nom de fichier dans la partie supérieure.
- Dans Connector Configurator, sélectionnez **File>Open**. Sélectionnez le nom du fichier de configuration du connecteur dans un projet ou dans le répertoire dans lequel il est stocké.
- Cliquez sur l'onglet Propriétés standard pour afficher les propriétés contenues dans ce fichier de configuration.

Création d'un modèle de propriétés spécifiques au connecteur

Pour créer un fichier de configuration pour votre connecteur, vous avez besoin d'un modèle de propriétés spécifiques au connecteur et des propriétés standard fournies par le système.

Vous pouvez créer un nouveau modèle pour les propriétés spécifiques au connecteur ou utiliser comme modèle une définition de connecteur existante.

- Pour créer un modèle, voir «Création d'un modèle», à la page 76.
- Pour utiliser un fichier existant, il vous suffit de modifier un modèle existant et de l'enregistrer sous le nouveau nom. Vous pouvez trouver des modèles existants dans le répertoire `\WebSphereAdapters\bin\Data\App`.

Création d'un modèle

Cette section décrit comment créer des propriétés dans le modèle, définir les valeurs et les caractéristiques générales de ces propriétés et indiquer toutes les dépendances entre les propriétés. Ensuite, vous pouvez utiliser le modèle comme base pour la création d'un nouveau fichier de configuration du connecteur.

Pour créer un modèle dans Connector Configurator, procédez comme suit :

1. Cliquez sur **File>New>Connector-Specific Property Template**.
2. La boîte de dialogue **Connector-Specific Property Template** s'affiche.
 - Entrez le nom du nouveau modèle dans la zone **Name** située sous **Input a New Template Name**. Vous voyez de nouveau ce nom lorsque vous ouvrez la boîte de dialogue pour créer un fichier de configuration à partir d'un modèle.
 - Pour afficher les définitions de propriétés spécifiques au connecteur dans n'importe quel modèle, sélectionnez le nom de ce modèle dans l'écran **Template Name**. La liste des définitions de propriétés contenues dans ce modèle apparaît dans l'écran **Template Preview**.
3. Vous pouvez utiliser un modèle existant dont les définitions de propriétés sont similaires à celles requises par votre connecteur comme point de départ pour votre modèle. Si aucun des modèles n'affiche les propriétés spécifiques au connecteur, vous devez en créer un.
 - Si vous prévoyez de modifier un modèle existant, sélectionnez le nom de ce modèle dans la liste située dans le tableau **Template Name** sous **Select the Existing Template to Modify: Find Template**.
 - Ce tableau affiche les noms de tous les modèles disponibles. Vous pouvez également rechercher un modèle.

Indication des caractéristiques générales

Lorsque vous cliquez sur **Next** pour sélectionner un modèle, la boîte de dialogue **Properties - Connector-Specific Property Template** s'affiche. Cette boîte de dialogue contient des onglets pour les caractéristiques générales des propriétés définies et pour les restrictions liées aux valeurs. L'écran général contient les zones suivantes :

- **General:**
 - Property Type
 - Property Subtype
 - Updated Method
 - Description
- **Flags**
 - Standard flags
- **Custom Flag**
 - Flag

Le **Property Subtype** peut être sélectionné si le **Property Type** est String. Cette valeur facultative entraîne la vérification de la syntaxe lors de l'enregistrement du fichier de configuration. La valeur par défaut est un espace, ce qui signifie que la propriété n'a reçu aucun sous-type.

Une fois que vous avez sélectionné les caractéristiques générales de la propriété, cliquez sur l'onglet **Value**.

Indication de valeurs

L'onglet **Value** vous permet de définir la longueur maximum, le nombre maximum de valeurs multiples, une valeur par défaut ou un intervalle de valeurs pour la propriété. Il autorise également les valeurs modifiables. Pour ce faire, procédez comme suit :

1. Cliquez sur l'onglet **Value**. Le panneau d'affichage des valeurs remplace le panneau d'affichage général.
2. Sélectionnez le nom de la propriété dans l'écran **Edit properties**.
3. Dans les zones relatives à la **longueur maximum** et au **nombre maximum de valeurs multiples**, entrez les valeurs de votre choix.

Pour créer une valeur de propriété, procédez comme suit :

1. Cliquez à l'aide du bouton droit de la souris sur le carré à gauche de l'en-tête de colonne **Value**.
2. Dans le menu en incrustation, sélectionnez **Add** pour afficher la boîte de dialogue **Property Value**. Selon le type de propriété, vous pourrez entrer une valeur avec ou sans un intervalle.
3. Entrez la nouvelle valeur de propriété et cliquez sur **OK**. La valeur apparaît dans le panneau **Value** situé dans la partie droite.

Le panneau **Value** contient un tableau comprenant trois colonnes :

La colonne **Value** contient la valeur que vous avez entrée dans la boîte de dialogue **Property Value** et toutes les valeurs que vous avez précédemment créées.

La colonne **Default Value** vous permet d'indiquer n'importe quelle valeur comme valeur par défaut.

La colonne **Value Range** contient l'intervalle que vous avez entré dans la boîte de dialogue **Property Value**.

Une fois que vous avez créé une valeur et qu'elle apparaît dans la grille, vous pouvez la modifier dans le tableau.

Pour modifier une valeur existante dans le tableau, sélectionnez une ligne entière en cliquant sur le numéro de ligne. Ensuite, cliquez avec le bouton droit dans la zone **Value** et cliquez sur **Edit Value**.

Définition des dépendances

Une fois les modifications apportées aux onglets **General** et **Value**, cliquez sur **Next**. La boîte de dialogue **Dependencies - Connector-Specific Property Template** s'affiche.

Une propriété dépendante est une propriété qui est incluse dans le modèle et utilisée dans le fichier de configuration *uniquement si* la valeur d'une autre propriété respecte une condition spécifique. Par exemple, `PollQuantity` apparaît dans le modèle uniquement si `JMS` est le mécanisme de transfert et que `DuplicateEventElimination` a la valeur `True`.

Pour faire en sorte qu'une propriété soit dépendante et définir la condition dont elle dépend, procédez comme suit :

1. Dans l'écran **Available Properties**, sélectionnez la propriété qui doit devenir dépendante.
2. Dans la zone **Select Property**, utilisez le menu déroulant pour sélectionner la propriété qui conservera la valeur conditionnelle.

3. Dans la zone **Condition Operator**, sélectionnez l'une des valeurs suivantes :
 - == (égal à)
 - != (différent de)
 - > (supérieur à)
 - < (inférieur à)
 - >= (supérieur ou égal à)
 - <= (inférieur ou égal à)
4. Dans la zone **Conditional Value**, entrez la valeur requise pour que la propriété dépendante soit incluse dans le modèle.
5. La propriété dépendante est mise en évidence dans l'écran **Available Properties** ; cliquez sur une flèche pour la déplacer vers l'écran **Dependent Property**.
6. Cliquez sur **Finish**. Connector Configurator stocke les informations que vous avez entrées sous la forme d'un document XML, sous \data\app dans le répertoire \bin où vous avez installé Connector Configurator.

Configuration des noms de chemins

Voici quelques règles générales pour la configuration des noms de chemins :

- Sous Windows et UNIX, un nom de fichier est limité à 255 caractères.
- Sous Windows, le nom de chemin absolu doit respecter le format [Unité:][Répertoire]\nom_de_fichier. Par exemple
C:\WebSphereAdapters\bin\Data\Std\StdConnProps.xml
Sous UNIX, le premier caractère doit être /.
- Un nom de file d'attente ne doit pas comporter d'espaces, ni à l'intérieur ni à la fin.

Création d'un fichier de configuration

Lorsque vous créez un fichier de configuration, vous devez lui attribuer un nom et sélectionner un courtier d'intégration.

Vous sélectionnez également un système d'exploitation pour effectuer une validation étendue du fichier. La barre d'outils dispose d'une liste déroulante nommée **Target System**, dans laquelle vous sélectionnez le système d'exploitation cible pour activer la validation étendue des propriétés. Les options sont : Windows, UNIX, Other (autres) et None (désactive la validation étendue). Au démarrage, la valeur par défaut est Windows.

Pour démarrer Connector Configurator :

- Dans la fenêtre System Manager, sélectionnez **Connector Configurator** dans le menu **Tools**. Connector Configurator s'ouvre.
- En mode autonome, démarrez Connector Configurator.

Pour définir le système d'exploitation et activer la validation étendue du fichier de configuration :

- Cliquez sur la liste déroulante **Target System**: de la barre de menus.
- Sélectionnez le système d'exploitation de votre environnement d'exécution.

Sélectionnez ensuite **File>New>Connector Configuration**. Dans la fenêtre New Connector, entrez le nom du nouveau connecteur.

Vous devez également sélectionner un courtier d'intégration. Le courtier que vous sélectionnez détermine les propriétés qui apparaîtront dans le fichier de configuration. Pour sélectionner un courtier, procédez comme suit :

- Dans la zone **Integration Broker**, sélectionnez ICS, les courtiers de messages WebSphere ou la connectivité WAS.
- Renseignez les zones restantes de la fenêtre **New Connector**, comme décrit précédemment dans ce chapitre.

Création d'un fichier de configuration pour un modèle spécifique au connecteur

Une fois que vous avez créé un modèle spécifique au connecteur, vous pouvez l'utiliser pour créer un fichier de configuration :

1. Indiquez le système d'exploitation pour la validation étendue du fichier de configuration, à l'aide de la liste déroulante **Target System**: de la barre de menus (voir ci-dessus "Création d'un nouveau fichier de configuration").
2. Cliquez sur **File>New>Connector Configuration**.
3. La boîte de dialogue **New Connector** contient les zones suivantes :

- **Name**

Entrez le nom du connecteur. Les noms font la différence entre les majuscules et les minuscules. Le nom que vous entrez doit être unique et cohérent avec le nom de fichier d'un connecteur installé sur le système.

Important : Connector Configurator ne contrôle pas l'orthographe du nom que vous entrez. Vous devez vérifier que le nom est correct.

- **System Connectivity**

Cliquez sur ICS, Courtiers de messages WebSphere ou WAS.

- **Select Connector-Specific Property Template**

Tapez le nom du modèle conçu pour votre connecteur. Les modèles disponibles s'affichent dans l'écran **Template Name**. Lorsque vous sélectionnez un nom dans l'écran **Template Name**, l'écran **Property Template Preview** affiche les propriétés spécifiques au connecteur qui ont été définies dans ce modèle.

Sélectionnez le modèle à utiliser et cliquez sur **OK**.

4. Un écran de configuration apparaît pour le connecteur que vous configurez. La barre de titre contient le nom du courtier d'intégration et du connecteur. Vous pouvez entrer les valeurs de toutes les zones pour terminer la définition maintenant, ou enregistrer le fichier et renseigner les zones ultérieurement.
5. Pour enregistrer le fichier, cliquez sur **File>Save>To File** ou sur **File>Save>To Project**. Pour exécuter un enregistrement dans un projet, System Manager doit être en cours d'exécution.

Si vous enregistrez un fichier, la boîte de dialogue **Save File Connector** apparaît. Sélectionnez *.cfg comme type de fichier, vérifiez dans la zone **File Name** que le nom est correctement orthographié et que sa casse est correcte, accédez au répertoire dans lequel vous souhaitez enregistrer le fichier et cliquez sur **Save**. L'écran d'état affiché dans le panneau de message de Connector Configurator indique que le fichier de configuration a été créé.

Important : Le nom et le chemin du répertoire que vous avez définis ici doivent correspondre au nom et au chemin du fichier de configuration du connecteur que vous indiquez dans le fichier de démarrage du connecteur.

6. Pour remplir la définition du connecteur, entrez des valeurs dans les zones de chacun des onglets de la fenêtre Connector Configurator, comme décrit plus loin dans ce chapitre.

Utilisation d'un fichier existant

Vous disposez peut-être d'un fichier existant dans un ou plusieurs des formats suivants :

- Fichier de définition du connecteur.
Il s'agit d'un fichier texte qui répertorie les propriétés et les valeurs par défaut applicables d'un connecteur spécifique. Certains connecteurs possèdent ce fichier dans un répertoire `\repository` fourni dans leur package d'origine (en général, le fichier a l'extension `.txt` ; par exemple, `CN_XML.txt` pour le connecteur XML).
- Fichier référentiel ICS.
Les définitions utilisées dans une implémentation ICS précédente du connecteur peuvent être accessibles dans un fichier référentiel qui a été utilisé pour la configuration de ce connecteur. En général, ce type de fichier a l'extension `.in` ou `.out`.
- Fichier de configuration précédent pour le connecteur.
En général, ce type de fichier a l'extension `*.cfg`.

Bien que certaines de ces sources de fichier puissent contenir tout ou partie des propriétés spécifiques au connecteur, le fichier de configuration du connecteur ne sera pas complet tant que vous n'aurez pas ouvert le fichier et défini les propriétés, comme décrit plus loin dans ce chapitre.

Pour utiliser un fichier existant afin de configurer un connecteur, vous devez ouvrir le fichier dans Connector Configurator, réviser la configuration et enregistrer de nouveau le fichier.

Pour ouvrir un fichier `*.txt`, `*.cfg` ou `*.in` dans un répertoire, procédez comme suit :

1. Dans Connector Configurator, cliquez sur **File>Open>From File**.
2. Dans la boîte de dialogue **Open File Connector**, sélectionnez l'un des types de fichier suivants pour afficher les fichiers disponibles :
 - Configuration (`*.cfg`)
 - Référentiel ICS (`*.in`, `*.out`)
Sélectionnez cette option si vous avez utilisé un fichier référentiel pour configurer le connecteur dans un environnement ICS. Un fichier référentiel peut contenir plusieurs définitions de connecteur, qui apparaissent toutes lorsque vous ouvrez ce fichier.
 - Tous les fichiers (`*.*`)
Sélectionnez cette option si un fichier `*.txt` a été fourni dans le package de l'adaptateur pour le connecteur ou qu'un fichier de définition avec une autre extension est disponible.
3. Dans l'écran du répertoire, accédez au fichier de définition du connecteur approprié, sélectionnez-le et cliquez sur **Open**.

Pour ouvrir une configuration de connecteur à partir d'un projet System Manager, procédez comme suit :

1. Démarrez System Manager. Vous pouvez ouvrir une configuration dans System Manager ou l'y enregistrer uniquement si vous avez démarré System Manager.
2. Démarrez Connector Configurator.

3. Cliquez sur **File>Open>From Project**.

Remplissage d'un fichier de configuration

Lorsque vous ouvrez un fichier de configuration ou un connecteur à partir d'un projet, la fenêtre Connector Configurator affiche l'écran de configuration qui contient les valeurs et les attributs courants.

Le titre de l'écran de configuration affiche le courtier d'intégration et le nom du connecteur, comme indiqué dans le fichier. Vérifiez que votre courtier est correct. Dans le cas contraire, modifiez la valeur du courtier avant de configurer le connecteur. Pour ce faire, procédez comme suit :

1. Dans l'onglet **Propriétés standard**, sélectionnez la zone de valeur pour la propriété BrokerType. Dans le menu déroulant, sélectionnez la valeur ICS, WMQI ou WAS.
2. L'onglet Standard Properties affiche les propriétés de connecteur associées au courtier sélectionné. La table indique les **Property name**, **Value**, **Type**, **Subtype** (si le Type est String), **Description** et **Update Method**.
3. Vous pouvez enregistrer le fichier maintenant ou renseigner les autres zones relatives à la configuration, comme décrit dans «Indication des définitions d'objets métier pris en charge», à la page 84.
4. Une fois la configuration terminée, cliquez sur **File>Save>To Project** ou sur **File>Save>To File**.

Si vous enregistrez dans un fichier, sélectionnez *.cfg comme extension, sélectionnez l'emplacement correct pour le fichier et cliquez sur **Save**.

Si plusieurs configurations de connecteur sont ouvertes, cliquez sur **Save All to File** pour enregistrer toutes les configurations dans un fichier ou cliquez sur **Save All to Project** pour enregistrer toutes les configurations du connecteur dans un projet System Manager.

Avant de créer le fichier de configuration, vous utiliserez la liste déroulante **Target System** pour sélectionner le système d'exploitation cible et activer la validation étendue des propriétés.

Avant d'enregistrer le fichier, Connector Configurator vérifie que vous avez défini des valeurs pour toutes les propriétés standard requises. Si vous n'avez pas défini de valeur pour l'une des propriétés standard requises, Connector Configurator affiche un message indiquant l'échec de la validation. Vous devez attribuer une valeur à la propriété pour pouvoir enregistrer le fichier de configuration.

Si vous avez activé la validation étendue en sélectionnant Windows, UNIX ou Other dans la liste déroulante **Target System**, le système validera les propriétés de type et de sous-type, et affichera un message en cas d'échec de la validation.

Définition des propriétés d'un fichier de configuration

Lorsque vous créez et que vous nommez un nouveau fichier de configuration du connecteur, ou que vous ouvrez un fichier de configuration existant du connecteur, Connector Configurator affiche un écran de configuration avec des onglets pour les catégories des valeurs de configuration requises.

Connector Configurator requiert des valeurs pour les propriétés dans ces catégories pour les connecteurs s'exécutant sur tous les courtiers :

- Propriétés standard
- Propriétés spécifiques au connecteur

- Objets métier pris en charge
- Valeurs des fichiers journaux/fichiers de trace
- Gestionnaire de données (applicable pour les connecteurs qui utilisent la messagerie JMS avec une livraison des événements garantie)

Remarque : Pour les connecteurs utilisant la messagerie JMS, une catégorie supplémentaire peut s'afficher ; elle est associée à la configuration des gestionnaires de données qui convertissent les données en objets métier.

Pour les connecteurs qui s'exécutent sur ICS, des valeurs sont également requises pour ces propriétés :

- Mappes associées
- Ressources
- Messagerie (le cas échéant)
- Sécurité

Important : Connector Configurator accepte que les valeurs des propriétés soient tapées en caractères anglais ou avec d'autres jeux de caractères. Cependant, les noms des propriétés standard et des propriétés spécifiques au connecteur ainsi que les noms des objets métier pris en charge doivent uniquement utiliser le jeu de caractères anglais.

Les différences entre les propriétés standard et les propriétés spécifiques au connecteur sont les suivants :

- Les propriétés standard d'un connecteur sont partagées par le composant spécifique à l'application d'un connecteur et son courtier. Tous les connecteurs ont le même jeu de propriétés standard. Ces propriétés sont décrites dans l'Annexe A de chaque guide de l'adaptateur. Vous pouvez modifier une partie de ces valeurs uniquement.
- Les propriétés spécifiques à l'application s'appliquent uniquement au composant spécifique à l'application d'un connecteur, c'est-à-dire au composant qui interagit directement avec l'application. Chaque connecteur a des propriétés spécifiques à l'application qui sont propres à cette application. Certaines de ces propriétés fournissent des valeurs par défaut, et d'autres non ; vous pouvez modifier certaines des valeurs par défaut. Les chapitres relatifs à l'installation et à la configuration de chaque guide de l'adaptateur décrivent les propriétés spécifiques à l'application et les valeurs recommandées.

Les zones relatives aux **propriétés standard** et aux **propriétés spécifiques au connecteur** sont codées en couleur pour indiquer les éléments configurables :

- Une zone avec un arrière-plan gris indique une propriété standard. Vous pouvez modifier la valeur, mais vous ne pouvez pas modifier le nom ou supprimer la propriété.
- Une zone avec un arrière-plan blanc indique une propriété spécifique à l'application. Ces propriétés varient en fonction des besoins spécifiques de l'application ou du connecteur. Vous pouvez modifier la valeur et supprimer ces propriétés.
- Les zones de valeurs sont configurables.
- La zone **Update Method** s'affiche pour chaque propriété. Elle indique si le redémarrage d'un composant ou d'un agent est nécessaire pour activer les valeurs modifiées. Vous ne pouvez pas configurer ce paramètre.

Définition des propriétés standard du connecteur

Pour modifier la valeur d'une propriété standard, procédez comme suit :

1. Cliquez dans la zone dont vous souhaitez définir la valeur.
2. Entrez une valeur ou sélectionnez-en une dans le menu déroulant le cas échéant.

Remarque : Si le Type de la propriété est String, la colonne Subtype peut contenir une valeur de sous-type. Ce sous-type sert pour la validation étendue de la propriété.

3. Une fois que vous avez entré toutes les valeurs pour les propriétés standard, vous pouvez exécuter les opérations suivantes :
 - Pour ignorer les modifications, conserver les valeurs d'origine et quitter Connector Configurator, cliquez sur **File>Exit** (ou fermez la fenêtre) et cliquez sur **No** lorsqu'un message vous demande si vous souhaitez enregistrer les modifications.
 - Pour entrer les valeurs des autres catégories dans Connector Configurator, sélectionnez l'onglet relatif à la catégorie. Les valeurs que vous entrez pour la catégories **Propriétés standard** (ou n'importe quelle autre catégorie) sont conservées lorsque vous passez à la catégorie suivante. Lorsque vous fermez la fenêtre, vous êtes invité à enregistrer ou à annuler les valeurs que vous avez entrées dans toutes les catégories.
 - Pour enregistrer les valeurs révisées, cliquez sur **File>Exit** (ou fermez la fenêtre) et sur **Yes** lorsqu'un message vous demande si vous souhaitez enregistrer les modifications. Vous pouvez également cliquer sur **Save>To File** dans le menu File ou la barre d'outils.

Pour plus d'informations sur une propriété standard donnée, cliquez sur l'entrée correspondante dans la colonne Description, dans la feuille à onglets Standard Properties. Si Extended Help est installé, un bouton flèche apparaît sur la droite. Un clic sur ce bouton ouvre une fenêtre d'aide, qui affiche des détails concernant la propriété standard.

Remarque : Si le bouton n'apparaît pas, c'est qu'il n'existe pas d'aide étendue pour cette propriété.

Les fichiers Extended Help sont installés dans le répertoire
<ProductDir>\bin\Data\Std\Help\<RegionalSetting>\.

Configuration des propriétés spécifiques au connecteur

Vous pouvez ajouter ou modifier des noms, définir des valeurs, supprimer une propriété spécifique ou la chiffrer. La longueur par défaut d'une propriété est de 255 caractères.

1. Cliquez avec le bouton droit dans la partie supérieure gauche de la grille. Une barre de menus contextuelle apparaît. Cliquez sur **Add** pour ajouter une propriété. Pour ajouter une propriété enfant, cliquez avec le bouton droit sur le numéro de la ligne parent et cliquez sur **Add child**.
2. Entrez une valeur pour la propriété ou la propriété enfant.

Remarque : Si la propriété est de Type String, vous pouvez sélectionner un sous-type dans la liste déroulante. Ce sous-type sert pour la validation étendue de la propriété.

3. Pour chiffrer une propriété, cochez la case **Encrypt**.

4. Pour plus d'informations sur une propriété donnée, cliquez sur l'entrée correspondante dans la colonne Description. Si Extended Help est installé, un bouton apparaît sur la droite. Un clic sur ce bouton ouvre une fenêtre d'aide, qui affiche des détails concernant la propriété.

Remarque : Si le bouton n'apparaît pas, c'est qu'il n'existe pas d'aide étendue pour cette propriété.

5. Vous pouvez enregistrer ou ignorer les modifications, comme décrit pour «Définition des propriétés standard du connecteur», à la page 83.

Si les fichiers Extended Help sont installés et que la propriété AdapterHelpName n'est pas renseignée, Connector Configurator pointera sur les fichiers Extended Help spécifiques au connecteur, dans le répertoire

`<ProductDir>\bin\Data\App\Help\<RegionalSetting>\`. Sinon, Connector Configurator pointera sur les fichiers Extended Help spécifiques au connecteur, dans le répertoire `<ProductDir>\bin\Data\App\Help\<AdapterHelpName>\<RegionalSetting>\`. Voir la propriété AdapterHelpName, décrite dans l'annexe Propriétés standard.

La zone Update Method affichée pour chaque propriété indique si le redémarrage d'un composant ou d'un agent est nécessaire à l'activation des valeurs modifiées.

Important : La modification du nom prédéfini d'une propriété de connecteur spécifique à l'application peut entraîner l'échec d'un connecteur. Le connecteur peut nécessiter certains noms de propriété pour se connecter à une application ou s'exécuter correctement.

Chiffrement des propriétés du connecteur

Pour chiffrer les propriétés spécifiques à l'application, cochez la case **Encrypt** dans la fenêtre des propriétés spécifiques au connecteur. Pour déchiffrer une valeur, décochez la case **Encrypt**, entrez la valeur appropriée dans la boîte de dialogue **Vérification** et cliquez sur **OK**. Si la valeur entrée est correcte, elle est déchiffrée et s'affiche.

Le guide d'utilisateur de l'adaptateur pour chaque connecteur contient la liste et la description de chaque propriété ainsi que sa valeur par défaut.

Si une propriété a plusieurs valeurs, la case **Encrypt** apparaît pour la première valeur de la propriété. Lorsque vous sélectionnez **Encrypt**, toutes les valeurs de la propriété sont chiffrées. Pour déchiffrer plusieurs valeurs d'une propriété, décochez la case **Encrypt** pour la première valeur de la propriété, puis entrez la nouvelle valeur dans la boîte de dialogue **Vérification**. Si la valeur entrée est une correspondance, toutes les valeurs multiples sont déchiffrées.

Méthode de mise à jour

Reportez-vous aux descriptions des méthodes de mise à jour, dans l'annexe Propriétés standard, sous «Présentation des valeurs des propriétés de configuration», à la page 48.

Indication des définitions d'objets métier pris en charge

Utilisez l'onglet **Supported Business Objects** dans Connector Configurator pour indiquer les objets métier que le connecteur utilisera. Vous devez indiquer les objets métier génériques et les objets métier spécifiques à l'application, et indiquer les associations pour les mappes entre les objets métier.

Remarque : Certains connecteurs nécessitent que des objets métier soient indiqués comme étant pris en charge pour pouvoir exécuter la notification des événements ou une configuration supplémentaire (à l'aide des méta-objets) avec leurs applications. Pour plus d'informations, voir *Connector Development Guide for C++* ou *Connector Development Guide for Java*.

Si ICS est votre courtier

Pour indiquer qu'une définition d'objet métier est prise en charge par le connecteur ou modifier les paramètres de prise en charge d'une définition d'objet métier existante, cliquez sur l'onglet **Supported Business Objects** et utilisez les zones suivantes :

Nom de l'objet métier : Pour indiquer qu'une définition d'objet métier est prise en charge par le connecteur, avec System Manager en cours d'exécution, procédez comme suit :

1. Cliquez dans une zone vide de la liste **Business Object Name**. Une liste déroulante s'affiche, avec toutes les définitions d'objet métier qui existent dans le projet System Manager.
2. Cliquez sur un objet métier pour l'ajouter.
3. Définissez la zone **Agent Support** (décrite plus bas) pour l'objet métier.
4. Dans le menu File de la fenêtre Connector Configurator, cliquez sur **Save to Project**. La définition révisée du connecteur, qui contient la prise en charge sélectionnée pour la définition de l'objet métier ajouté, est enregistrée dans un projet ICL (Integration Component Library) de System Manager.

Pour supprimer un objet métier dans la liste des objets métier pris en charge :

1. Pour sélectionner la zone d'un objet métier, cliquez sur le numéro situé à gauche de l'objet métier.
2. Dans le menu **Edit** de la fenêtre Connector Configurator, cliquez sur **Delete Row**. L'objet métier est supprimé de la liste.
3. Dans le menu **File**, cliquez sur **Save to Project**.

La suppression d'un objet métier dans la liste des objets métier pris en charge modifie la définition du connecteur et rend l'objet métier supprimé inutilisable dans cette implémentation du connecteur. Elle n'affecte pas le code du connecteur et ne supprime pas la définition de l'objet métier dans System Manager.

Prise en charge de l'agent : Si un objet métier dispose de la prise en charge de l'agent, le système tente d'utiliser cet objet métier pour fournir des données à une application via l'agent du connecteur.

En général, les objets métier spécifiques à l'application pour un connecteur sont pris en charge par l'agent de ce connecteur, mais les objets métier génériques ne le sont pas.

Pour indiquer que l'objet métier est pris en charge par l'agent du connecteur, cochez la case **Agent Support**. La fenêtre Connector Configurator ne valide pas vos sélections pour Agent Support.

Niveau de transaction maximum : Le niveau de transaction maximum d'un connecteur correspond au niveau de transaction le plus élevé pris en charge par le connecteur.

Pour la plupart des connecteurs, Best Effort est la seule valeur possible.

Vous devez redémarrer le serveur pour que les modifications prennent effet.

Si votre courtier est un courtier de messages WebSphere

Si vous utilisez le mode autonome (non connecté à System Manager), vous devez entrer manuellement le nom de l'objet métier.

Si System Manager est en cours d'exécution, vous pouvez cocher la case située sous la colonne **Business Object Name** dans l'onglet **Supported Business Objects**. Une boîte de dialogue mixte affiche la liste des objets métier disponibles dans le projet Integration Component Library auquel le connecteur appartient. Dans cette liste, sélectionnez l'objet métier de votre choix.

La zone **Message Set ID** est facultative pour WebSphere Business Integration Message Broker 5.0, et sa valeur ne doit pas nécessairement être unique le cas échéant. Cependant, pour WebSphere MQ Integrator et Integrator Broker 2.1, vous devez indiquer un **ID** unique.

Si WAS est votre courtier

Lorsque vous sélectionnez WebSphere Application Server comme type de courtier, Connector Configurator ne nécessite pas les ID d'ensemble de messages. L'onglet **Supported Business Objects** contient la colonne **Business Object Name** pour les objets métier pris en charge uniquement.

Si vous utilisez le mode autonome (non connecté à System Manager), vous devez entrer manuellement le nom de l'objet métier.

Si System Manager est en cours d'exécution, vous pouvez cocher la case située sous la colonne Business Object Name dans l'onglet Supported Business Objects. Une boîte de dialogue mixte affiche la liste des objets métier disponibles dans le projet Integration Component Library auquel le connecteur appartient. Dans cette liste, sélectionnez l'objet métier de votre choix.

Mappes associées (ICS)

Chaque connecteur prend en charge la liste des définitions des objets métier et leurs mappes associées actives dans WebSphere InterChange Server. Cette liste apparaît lorsque vous sélectionnez l'onglet **Associated Maps**.

La liste des objets métier contient l'objet métier spécifique à l'application pris en charge par l'agent et l'objet générique correspondant que le contrôleur envoie à la collaboration de souscription. L'association d'une mappe détermine la mappe qui sera utilisée pour transformer l'objet métier spécifique à l'application en objet métier générique, ou inversement.

Si vous utilisez des mappes uniquement définies pour des objets métier source et cible spécifiques, les mappes sont déjà associées aux objets métier appropriés lorsque vous affichez l'écran, et vous n'avez pas besoin de (ou ne pouvez pas) les modifier.

Si plusieurs mappes sont disponibles pour un objet métier pris en charge, vous devez lier de manière explicite cet objet métier à la mappe qu'il doit utiliser.

L'onglet **Associated Maps** affiche les zones suivantes :

- **Business Object Name**

Il s'agit des objets métier pris en charge par ce connecteur, comme indiqué dans l'onglet **Supported Business Objects**. Si vous indiquez des objets métier

supplémentaires dans l'onglet Supported Business Objects, ils sont reflétés dans cette liste une fois que vous avez enregistré les modifications en sélectionnant **Save to Project** dans le menu **File** de la fenêtre Connector Configurator.

- **Associated Maps**

L'écran affiche toutes les cartes installées sur le système à utiliser avec les objets métier pris en charge du connecteur. L'objet métier source pour chaque carte s'affiche à gauche du nom de la carte, dans l'écran **Business Object Name**.

- **Liaison explicite**

Dans certains cas, vous devrez peut-être lier de manière explicite une carte associée.

Une liaison explicite est requise uniquement lorsque plusieurs cartes existent pour un objet métier pris en charge spécifique. Lorsque ICS s'amorce, il tente de lier automatiquement une carte à chaque objet métier pris en charge pour chacun des connecteurs. Si plusieurs cartes prennent le même objet métier comme entrée, le serveur tente de localiser et de lier une carte qui correspond au sur-ensemble des autres.

Si aucune carte n'est le sur-ensemble des autres, le serveur ne peut pas lier l'objet métier à une seule carte et vous devrez définir la liaison de manière explicite.

Pour lier une carte de manière explicite, procédez comme suit :

1. Dans la colonne **Explicit**, cochez la case correspondant à la carte à lier.
2. Sélectionnez la carte que vous souhaitez associer à l'objet métier.
3. Dans le menu **File** de la fenêtre Connector Configurator, cliquez sur **Save to Project**.
4. Déployez le projet jusqu'à ICS.
5. Réamorcez le serveur pour que les modifications prennent effet.

Ressources (ICS)

L'onglet **Resource** vous permet de définir une valeur qui détermine si l'agent du connecteur gèrera plusieurs processus simultanément, et dans quelle mesure, à l'aide du parallélisme de l'agent du connecteur.

Tous les connecteurs ne prennent pas en charge cette fonction. Si vous exécutez un agent de connecteur conçu dans Java pour être multithread, nous vous recommandons de ne pas utiliser cette fonction dans la mesure où il est généralement plus efficace d'utiliser plusieurs unités d'exécution plutôt que plusieurs processus.

Messagerie (ICS)

L'onglet **Messaging** vous permet de configurer les propriétés de messagerie. Les propriétés de messagerie sont disponibles uniquement si vous avez défini MQ comme la valeur de la propriété standard `DeliveryTransport` et ICS comme le type de courtier. Ces propriétés affectent la manière dont le connecteur utilisera les files d'attente.

Validation des files d'attente de messages

Avant de valider une file d'attente de messages, vous devez :

- Vous assurer que WebSphere MQ Series est installé.
- Créer sur la machine hôte une file d'attente de messages avec le canal et le port.
- Configurer une connexion sur la machine hôte.

Pour valider la file d'attente, vous utiliserez le bouton Valider à droite des zones Messaging Type et Host Name, dans l'onglet Messaging.

Sécurité (ICS)

L'onglet **Security** du Connector Configurator permet de définir le niveau de confidentialité d'un message. Cette propriété n'est utilisable que si la propriété DeliveryTransport est définie sur JMS.

Par défaut, Privacy est désactivé. Pour l'activer, cochez la case **Privacy**.

Le **Keystore Target System Absolute Pathname** (Chemin absolu du magasin de clés du système cible) est :

- Pour Windows :
<ProductDir>\connectors\security\- pour UNIX :
opt/IBM/WebSphereAdapters/connectors/security/<connectorname>.jks

Ce chemin et le fichier qu'il indique doivent être sur le système qui vous servira à démarrer le connecteur, c'est-à-dire le système cible.

Vous pouvez utiliser le bouton Browse à droite, à condition que le système cible soit le seul en cours de fonctionnement. Ce bouton est désactivé jusqu'à ce que **Privacy** soit activé et que le **Target System** de la barre de menus soit défini sur Windows.

Le **Message Privacy Level** peut être défini comme suit pour les trois catégories de messages (All Messages, All Administrative Messages, et All Business Object Messages) :

- La valeur par défaut "" indique qu'aucun niveau de confidentialité n'a été défini pour une catégorie de messages.
- none
Cette valeur n'a pas le même sens que la valeur par défaut : elle indique le choix délibéré d'attribuer ce niveau de confidentialité à une catégorie de messages.
- integrity
- privacy
- integrity_plus_privacy

La fonction **Key Maintenance** permet de générer, importer et exporter des clés publiques pour le serveur et l'adaptateur.

- La sélection de **Generate Keys** ouvre la boîte de dialogue correspondante, avec les valeurs par défaut pour l'outil qui générera les clés.
- Le magasin de clés est par défaut celui que vous avez indiqué pour **Keystore Target System Absolute Pathname** dans l'onglet Security.
- Lorsque vous cliquez sur OK, les entrées sont validées, le certificat est généré et la sortie est envoyée à la fenêtre de connexion du Connector Configurator.

Avant d'importer un certificat dans le magasin de clés de l'adaptateur, vous devez l'exporter depuis le magasin de clés du serveur. Pour cela, sélectionnez **Export Adapter Public Key**, ce qui ouvre la fenêtre correspondante.

- Par défaut, le certificat exporté prend les valeurs du magasin de clés, à l'exception de son extension qui est <nomdefichier>.cer.

Lorsque vous sélectionnez **Import Server Public Key**, la boîte de dialogue correspondante s'ouvre.

- Par défaut, le certificat importé prend la valeur `<ProductDir>\bin\ics.cer` (si le fichier existe sur le système).
- Le nom de serveur doit être la Certificate Association d'importation. Si un serveur est enregistré, vous pouvez le sélectionner dans la liste déroulante.

La fonction **Adapter Access Control** n'est activée que si `DeliveryTransport` est définie sur `IDL`. Par défaut l'adaptateur se connecte avec l'identité d'invité (guest). Si la case **Use guest identity** n'est pas cochée, les zones **Adapter Identity** et **Adapter Password** sont accessibles.

Définition des valeurs du fichier de trace ou du fichier journal

Lorsque vous ouvrez le fichier de configuration ou le fichier de définitions d'un connecteur, Connector Configurator utilise les valeurs de journalisation et de trace de ce fichier comme valeurs par défaut. Vous pouvez modifier ces valeurs dans Connector Configurator.

Pour modifier les valeurs de journalisation et de trace, procédez comme suit :

1. Cliquez sur l'onglet **Trace/Log Files**.
2. Pour la journalisation ou la fonction de trace, vous pouvez écrire des messages à l'un des composants suivants :
 - A la console (STDOUT) :
Ecrit des messages de journalisation ou de trace à l'écran STDOUT.

Remarque : Vous pouvez utiliser l'option `STDOUT` de l'onglet **Trace/Log Files** pour les connecteurs s'exécutant sur la plate-forme Windows.

- A un fichier :
Ecrit des messages de journalisation ou de trace vers un fichier indiqué. Pour indiquer le fichier, cliquez sur le bouton du répertoire (ellipse), accédez à l'emplacement de votre choix, indiquez un nom de fichier et cliquez sur **Save**. Les messages de journalisation ou de trace sont écrits vers le fichier et l'emplacement indiqués.

Remarque : Les fichiers de journalisation et de trace sont de simples fichiers texte. Vous pouvez utiliser l'extension de votre choix lorsque vous définissez les noms de fichier. Cependant, pour les fichiers de trace, nous vous recommandons d'utiliser l'extension `.trace` plutôt que l'extension `.trc`, afin d'éviter toute confusion avec les autres fichiers pouvant résider sur le système. Pour les fichiers de journalisation, les extensions classiques sont `.log` et `.txt`.

Gestionnaires de données

La section des gestionnaires de données est disponible pour la configuration uniquement si vous avez indiqué une valeur `JMS` pour `ContainerManagedEvents`. Tous les adaptateurs n'utilisent pas les gestionnaires de données.

Pour connaître les valeurs à utiliser pour ces propriétés, reportez-vous aux descriptions sous `ContainerManagedEvents` dans l'Annexe A, Propriétés standard. Pour plus d'informations, voir *Connector Development Guide for C++* ou *Connector Development Guide for Java*.

Enregistrement de votre fichier de configuration

Une fois que vous avez configuré votre connecteur, enregistrez son fichier de configuration. Connector Configurator enregistre le fichier dans le mode courtier que vous avez sélectionné pendant la configuration. La barre de titre de Connector Configurator affiche toujours le mode courtier (ICS, WMQI ou WAS) en cours d'utilisation.

Le fichier est enregistré en tant que document XML. Pour enregistrer le document XML, vous avez trois possibilités :

- dans System Manager, en tant que fichier avec l'extension *.con dans le projet ICL ;
- dans un répertoire que vous avez indiqué ;
- en mode autonome, en tant que fichier avec l'extension *.cfg dans un répertoire (par défaut, le fichier est enregistré dans \WebSphereAdapters\bin\Data\App) ;
- dans un projet WebSphere Application Server, le cas échéant.

Pour plus d'informations sur l'utilisation des projets dans System Manager et sur le déploiement, voir les guides d'implémentation suivants :

- Pour ICS : *Implementation Guide for WebSphere InterChange Server*
- Pour les courtiers de messages WebSphere : *Implementing Adapters with WebSphere Message Brokers*
- Pour WAS : *Implementing Adapters with WebSphere Application Server*

Modification d'un fichier de configuration

Vous pouvez modifier les paramètres du courtier d'intégration pour un fichier de configuration existant. Cela vous permet d'utiliser le fichier comme modèle pour la création d'un nouveau fichier de configuration que vous pouvez utiliser avec un autre courtier.

Remarque : Vous devrez modifier d'autres propriétés de configuration ainsi que la propriété du mode courtier si vous changez de courtiers d'intégration.

Pour modifier votre sélection de courtier dans un fichier de configuration existant (facultatif) :

- Ouvrez le fichier de configuration existant dans Connector Configurator.
- Sélectionnez l'onglet **Standard Properties**.
- Dans la zone **BrokerType** de l'onglet Standard Properties, sélectionnez la valeur appropriée pour votre courtier.
Lorsque vous modifiez la valeur courante, les onglets disponibles et les sélections de zones de la fenêtre des propriétés changent immédiatement, pour ne montrer que les onglets et zones qui correspondent au courtier que vous avez sélectionné.

Exécution de la configuration

Une fois que vous avez créé un fichier de configuration pour un connecteur et que vous l'avez modifié, assurez-vous que le connecteur peut localiser le fichier de configuration lorsqu'il démarre.

Pour ce faire, ouvrez le fichier de démarrage utilisé pour le connecteur et vérifiez que le nom de fichier et l'emplacement utilisés pour le fichier de configuration du connecteur correspondent exactement au nom attribué au fichier et au répertoire ou au chemin d'accès dans lequel vous l'avez placé.

Utilisation de Connector Configurator dans un environnement globalisé

Connector Configurator est globalisé et peut gérer la conversion des caractères entre le fichier de configuration et le courtier d'intégration. Connector Configurator utilise le codage natif. Lorsqu'il écrit dans le fichier de configuration, il utilise le codage UTF-8.

Connector Configurator prend en charge les caractères qui n'existent pas en anglais dans :

- toutes les zones de valeur ;
- le chemin d'accès au fichier journal et au fichier de trace (indiqué dans l'onglet **Trace/Log files**).

La liste déroulante pour les propriétés de configuration standard CharacterEncoding et Locale affiche uniquement un sous-ensemble des valeurs prises en charge. Pour ajouter d'autres valeurs à cette liste, vous devez modifier manuellement le fichier \Data\Std\stdConnProps.xml dans le répertoire produit.

Par exemple, pour ajouter l'environnement local en_GB à la liste des valeurs pour la propriété Locale, ouvrez le fichier stdConnProps.xml et ajoutez la ligne en caractère gras comme indiqué ci-dessous :

```
<Property name="Locale"
isRequired="true"
updateMethod="component restart">
  <ValidType>String</ValidType>
  <ValidValues>
    <Value>ja_JP</Value>
    <Value>ko_KR</Value>
    <Value>zh_CN</Value>
    <Value>zh_TW</Value>
    <Value>fr_FR</Value>
    <Value>de_DE</Value>
    <Value>it_IT</Value>
    <Value>es_ES</Value>
    <Value>pt_BR</Value>
    <Value>en_US</Value>
    <Value>en_GB</Value>
  </ValidValues>
  <DefaultValue>en_US</DefaultValue>
</Property>
```

Annexe C. Application Response Measurement

Cet adaptateur est compatible avec l'API Application Response Measurement, qui permet de gérer la disponibilité, les accords de niveau de service et la planification de la capacité des applications. Une application avec appels ARM peut participer à IBM Tivoli Monitoring for Transaction Performance, autorisant ainsi la collecte et la consultation des données relatives aux indicateurs de transaction.

Prise en charge des appels Application Response Measurement

Cet adaptateur est compatible avec l'API Application Response Measurement, qui permet de gérer la disponibilité, les accords de niveau de service et la planification de la capacité des applications. Une application avec appels ARM peut participer à IBM Tivoli Monitoring for Transaction Performance, autorisant ainsi la collecte et la consultation des données relatives aux indicateurs de transaction.

Logiciels requis

En plus des logiciels exigés par l'adaptateur, les applications suivantes doivent être installées pour qu'ARM puisse fonctionner :

- WebSphere Application Server 5.0.1 (contient le serveur IBM Tivoli Monitoring for Transaction Performance). Il ne doit pas forcément être installé sur le même système que l'adaptateur.
- IBM Tivoli Monitoring for Transaction Performance v. 5.2 Fixpack 1. Il doit être installé sur le même système que l'adaptateur, et configuré pour pointer vers la machine qui héberge le serveur IBM Tivoli Monitoring for Transaction Performance.

La prise en charge de Application Response Measurement est possible avec tout courtier d'intégration supporté par cette édition.

Remarque : Les appels Application Response Measurement sont pris en charge par tous les systèmes d'exploitation supportés par la présente édition de IBM WebSphere Business Integration Adapters, *sauf* par HP-UX (toutes versions) et Red Hat Linux 3.0.

Activation de Application Response Measurement

Les appels ARM sont activés en définissant sur "True" la propriété standard `TivoliMonitorTransactionPerformance` de Connector Configurator. Par défaut, la prise en charge ARM est désactivée. (Pour plus d'informations, voir l'annexe "Propriétés standard" de ce document.)

Surveillance des transactions

Lorsque ARM est activé, les transactions surveillées sont les événements de service et les livraisons d'événements. La transaction est mesurée à partir du début de la demande de service ou de la livraison d'événement et jusqu'à la fin. Le nom de la transaction, affiché sur la console Tivoli Monitoring for Transaction Performance, commencera par SERVICE REQUEST ou EVENT DELIVERY. La suite du nom sera l'instruction de l'objet métier (comme CREATE, RETRIEVE, UPDATE ou DELETE). La fin du nom sera le nom de l'objet métier, comme "EMPLOYEE." Par exemple, le nom d'une transaction pour une livraison d'événement correspondant à la création d'un employé pourra être EVENT DELIVERY CREATE EMPLOYEE ou SERVICE REQUEST UPDATE ORDER.

Les indicateurs ci-après sont collectés par défaut pour chaque type de demande de service ou de livraison d'événement :

- Durée minimale de transaction
- Durée maximale de transaction
- Durée moyenne de transaction
- Total d'exécutions de la transaction

Vous (ou l'administrateur système de WebSphere Application Server) pouvez sélectionner les indicateurs à afficher, et pour quels événements d'adaptateur, en configurant Discovery Policies et Listener Policies pour des transactions données, depuis la console Tivoli Monitoring for Transaction Performance. (Voir «Pour plus d'informations».)

Pour plus d'informations

Pour plus de détails, voir la documentation IBM Tivoli Monitoring for Transaction Performance. Pour obtenir des informations sur la surveillance et la gestion des indicateurs générés par l'adaptateur, voir *IBM Tivoli Monitoring for Transaction Performance User's Guide*.

Annexe D. Common Event Infrastructure

WebSphere Business Integration Server Foundation inclut Common Event Infrastructure Server Application, nécessaire au fonctionnement de Common Event Infrastructure. WebSphere Application Server Foundation peut être installé sur tout système (pas nécessairement sur la même machine que l'adaptateur.)

WebSphere Application Server Application Client contient les bibliothèques nécessaires à l'interaction entre l'adaptateur et Common Event Infrastructure Server Application. Vous devez installer WebSphere Application Server Application Client sur le même système que l'adaptateur. L'adaptateur se connecte à WebSphere Application Server (dans WebSphere Business Integration Server Foundation) par le biais d'une URL configurable.

La prise en charge de Common Event Infrastructure est possible avec tout courtier d'intégration supporté par cette édition.

Logiciels requis

En plus des logiciels exigés pour l'adaptateur, les applications suivantes doivent être installées pour que Common Event Infrastructure puisse fonctionner :

- WebSphere Business Integration Server Foundation 5.1.1
- WebSphere Application Server Application Client 5.0.2, 5.1 ou 5.1.1.
(WebSphere Application Server Application Client 5.1.1 est fourni avec WebSphere Business Integration Server Foundation 5.1.1.)

Remarque : Common Event Infrastructure n'est pris en charge par aucune plateforme HP-UX ou Linux.

Activation de Common Event Infrastructure

La fonctionnalité Common Event Infrastructure est activée avec les propriétés standard `CommonEventInfrastructure` et `CommonEventInfrastructureContextURL`, configurées avec Connector Configurator. Par défaut, Common Event Infrastructure n'est pas activé. La propriété `CommonEventInfrastructureContextURL` vous permet de configurer l'URL du serveur Common Event Infrastructure. (Pour plus d'informations, voir l'annexe "Propriétés standard" du présent document.)

Obtention d'événements d'adaptateur Common Event Infrastructure

Si Common Event Infrastructure est activé, l'adaptateur génère des événements Common Event Infrastructure qui se mappent sur les événements d'adaptateur suivants :

- Démarrage de l'adaptateur
- Arrêt de l'adaptateur
- Réponse de l'application à un dépassement de délai de l'agent de l'adaptateur
- Tout appel `doVerbFor` émis depuis l'agent de l'adaptateur
- Appel `gotAppEvent` depuis l'agent de l'adaptateur

Pour qu'une autre application (l'"application client") puisse recevoir les événements Common Event Infrastructure générés par l'adaptateur, elle doit

utiliser le catalogue d'événements Common Event Infrastructure pour déterminer les définitions des événements appropriés et leurs propriétés. Pour que l'application client puisse utiliser les événements de l'application d'origine, les événements doivent être définis dans le catalogue d'événements.

L'annexe "Définitions du catalogue des événements Common Event Infrastructure" du présent document contient des métadonnées au format XML. Elles indiquent, pour les adaptateurs WebSphere Business Information, les descripteurs d'événements et les propriétés que l'application client doit rechercher.

Pour plus d'informations

Pour plus d'informations sur Common Event Infrastructure, voir la documentation WebSphere Business Integration Server Foundation, disponible à l'adresse :

<http://publib.boulder.ibm.com/infocenter/ws51help>

Pour consulter des exemples de métadonnées XML indiquant les propriétés et descripteurs d'événement générés par l'adaptateur qu'une application client doit utiliser, voir «Définitions du catalogue d'événements Common Event Infrastructure».

Définitions du catalogue d'événements Common Event Infrastructure

Le catalogue d'événements Common Event Infrastructure contient des définitions d'événements qui peuvent être interrogées par d'autres applications. Ci-dessous figurent des exemples de définitions d'événements utilisant des métadonnées XML pour des événements d'adaptateur types. Si vous écrivez une autre application, elle peut utiliser les interfaces de catalogue d'événements pour interroger la définition de l'événement. Pour plus d'informations sur les définitions d'événements et savoir comment les interroger, voir la documentation Common Event Infrastructure disponible dans le centre d'information en ligne IBM WebSphere Server Foundation.

Pour les adaptateurs WebSphere Business Integration, les éléments de données étendues qui doivent être définis dans le catalogue des événements sont les clés de l'objet métier. Chaque clé d'objet métier exige une définition d'événement. Par conséquent, pour un adaptateur donné, les divers événements tels que start adapter, stop adapter, timeout adapter et tout événementdoVerbFor (comme create, update ou delete) doivent avoir une définition dans le catalogue d'événements.

Les sections suivantes contiennent des exemples des métadonnées pour start adapter, stop adapter et la requête ou la livraison d'événement.

Format XML des métadonnées de "start adapter"

```
<eventDefinition name="startADAPTER"
  parent="event">
  <property name="creationTime" //Commentaire : Par exemple, la valeur peut être
    "2004-05-13T17:00:16.319Z"
    required="true" />
  <property name="globalInstanceId" //Commentaire : Généré automatiquement
    par Common Event Infrastructure
    required="true"/>
  <property name="sequenceNumber" //Commentaire : Numéro défini par la source
    pour les messages qui seront envoyés/triés de façon logique
    required="false"/>
  <property name="version" //Commentaire : Version de l'événement
```



```

        required="false"
        defaultValue="1.0.1"/>
    <property name="sourceComponentId"
        path="sourceComponentId"
        required="true"/>
    <property name="application" //Commentaire : Le name#version de
l'application source générant l'événement. Par exemple, "SampleConnector#3.0.0"
        path="sourceComponentId/application" required="false"/>
    <property name="component" //Commentaire : Ce sera le name#version
du composant source.
        path="sourceComponentId/component"
        required="true"
        defaultValue="ConnectorFrameWorkVersion#4.2.2"/>
    <property name="componentIdType" //Commentaire : Précise le format
et la signification du composant
        path="sourceComponentId/componentIdType"
        required="true"
        defaultValue="Application"/>
    <property name="executionEnvironment"
//Commentaire : Identifie l'environnement dans lequel l'application est exécutée
...par exemple "Windows 2000#5.0"
        path="sourceComponentId/executionEnvironment"
        required="false" />
    <property name="location" //Commentaire : Cette valeur correspond au
nom du serveur... par exemple "WQMI"
        path="sourceComponentId/location"
        required="true"/>
    <property name="locationType" //Commentaire : Indique le format et
la signification de l'emplacement
        path="sourceComponentId/locationType"
        required="true"
        defaultValue="Hostname"/>
    <property name="subComponent" //Commentaire : Distinction supplémentaire
du composant logique
        path="sourceComponentId/subComponent"
        required="true"
        defaultValue="AppSide_Connector.AgentBusinessObjectManager"/>
    <property name="componentType" //Commentaire : Nom correctement défini
utilisé pour caractériser toutes les instances de ce composant
        path="sourceComponentId/componentType" required="true"
        defaultValue="ADAPTER"/>
    <property name="situation" //Commentaire : Définit le type de
situation ayant entraîné le signalement de l'événement
        path="situation"
        required="true"/>
    <property name="categoryName=" //Commentaire : Indique le type
de situation pour l'événement
        path="situation/categoryName"
        required="true"
        defaultValue="StartSituation"/>
    <property name="situationType" //Commentaire : Indique le type
de situation et la disposition de l'événement
        path="situation/situationType"
        required="true"
    <property name="reasoningScope" //Commentaire : Indique
l'impact de l'événement
        path="situation/situationType/reasoningScope"
        required="true"
        permittedValue="INTERNAL"
        permittedValue="EXTERNAL"/>
    <property name="successDisposition" //Commentaire : Indique la
réussite de l'événement
        path="situation/situationType/successDisposition"
        required="true"
        permittedValue="SUCCESSFUL"
        permittedValue="UNSUCCESSFUL" />
    <property name="situationQualifier" //Commentaire : Indique les

```

```

qualificatifs de la situation pour cet événement
    path="situation/situationType/situationQualifier"
    required="true"
    permittedValue="START_INITIATED"
    permittedValue="RESTART_INITIATED"
    permittedValue="START_COMPLETED" />
</eventDefinition>

```

Format XML des métadonnées de "stop adapter"

Les métadonnées de "stop adapter" sont identiques à celles de "start adapter" aux exceptions suivantes près :

- La valeur par défaut de la propriété categoryName est StopSituation :

```

<property name="categoryName="
  //Commentaire : Indique le type
  de situation pour l'événement
    path="situation/categoryName"
    required="true"
    defaultValue="StopSituation"/>

```

- Les valeurs autorisées pour la propriété situationQualifier sont différentes et sont les suivantes pour "stop adapter" :

```

<property name="situationQualifier"
  //Commentaire : Précise les qualificatifs de la situation pour cet événement
    path="situation/situationType/situationQualifier"
    required="true"
    permittedValue="STOP_INITIATED"
    permittedValue="ABORT_INITIATED"
    permittedValue="PAUSE_INITIATED"
    permittedValue="STOP_COMPLETED"
  />

```

Format XML des métadonnées de "timeout adapter"

Les métadonnées de "timeout adapter" sont identiques à celles de "start adapter" et "stop adapter" aux exceptions suivantes près :

- La valeur par défaut de la propriété categoryName est ConnectSituation :

```

<property name="categoryName="
  //Commentaire : Indique le type
  de situation pour l'événement
    path="situation/categoryName"
    required="true"
    defaultValue="ConnectSituation"/>

```

- Les valeurs autorisées pour la propriété situationQualifier sont différentes et sont les suivantes pour "timeout adapter" :

```

<property name="situationQualifier" //Commentaire : Précise
  les qualificatifs de la situation pour cet événement
    path="situation/situationType/situationQualifier"
    required="true"
    permittedValue="IN_USE"
    permittedValue="FREED"
    permittedValue="CLOSED"
    permittedValue="AVAILABLE"
  />

```

Format XML des métadonnées de "request" ou "delivery"

A la fin de ce format XML figurent les éléments de données étendues. Les éléments de données étendues des événements de livraison et de requête d'adaptateur sont les données de l'objet métier en cours de traitement. Ces données incluent le nom de l'objet métier, sa clé (étrangère ou locale) et les objets métier qui sont des enfants d'objets métier parents. Les objets métier enfants sont ensuite décomposés dans les mêmes données que le parent (nom, clé et tout objet métier enfant). Ces données sont représentées dans un élément de données étendues de la définition de l'événement. Ces données changeront selon l'objet métier, les clés et les objets métier enfants traités. Les données étendues de cette définition d'événement sont un simple exemple et représentent un objet métier nommé Employee avec une clé EmployeeId et un objet métier enfant EmployeeAddress avec une clé EmployeeId. Ce modèle peut s'appliquer à toutes les données d'un objet métier particulier.

```
<eventDefinition name="createEmployee" //Commentaire : Ce
nom d'extension est toujours l'instruction de l'objet métier suivi de
son nom
  parent="event">
  <property name="creationTime" //Commentaire : Par exemple, la valeur peut être
"2004-05-13T17:00:16.319Z"
    required="true" />
  <property name="globalInstanceId" //Commentaire : Généré automatiquement
par Common Event Infrastructure
    required="true"/>
  <property name="localInstanceId" //Commentaire : La valeur est égale à
l'instruction de l'objet métier+nom de l'objet métier+#+nom app+ identificateur
de l'objet métier
    required="false"/>
  <property name="sequenceNumber" //Commentaire : Numéro défini par la source
pour les messages qui seront envoyés/triés de façon logique
    required="false"/>
  <property name="version" //Commentaire : Version de l'événement... la valeur
est définie sur 1.0.1
    required="false"
    defaultValue="1.0.1"/>
  <property name="sourceComponentId"
    path="sourceComponentId"
    required="true"/>
  <property name="application" //Commentaire : Le name#version de
l'application source qui génère l'événement... par exemple
"SampleConnector#3.0.0"
    path="sourceComponentId/application"
    required="false"/>
  <property name="component" //Commentaire : Ce sera le name#version
du composant source.
    path="sourceComponentId/component"
    required="true"
    defaultValue="ConnectorFrameWorkVersion#4.2.2"/>
  <property name="componentIdType" //Commentaire : Précise le format
et la signification du composant
    path="sourceComponentId/componentIdType"
    required="true"
    defaultValue="Application"/>
  <property name="executionEnvironment" //Commentaire : Identifie le
environment#version dans lequel l'application est exécutée... par exemple
"Windows 2000#5.0"
    path="sourceComponentId/executionEnvironment"
    required="false" />
  <property name="instanceId" //Commentaire : La valeur est égale à l'instruction
de l'objet métier+nom de l'objet métier+#+nom app+identificateur de l'objet métier
    path="sourceComponentId/instanceId"
    required="false"
  <property name="location" //Commentaire : Cette valeur correspond au
nom du serveur... par exemple "WQMI"
```

```

        path="sourceComponentId/location"
        required="true"/>
    <property name="locationType" //Commentaire : Indique le format et
    la signification de l'emplacement
        path="sourceComponentId/locationType"
        required="true"
        defaultValue="Hostname"/>
    <property name="subComponent" //Commentaire : Distinction supplémentaire du
    composant logique (dans ce cas, la valeur est égale au nom de l'objet
    métier)
        path="sourceComponentId/subComponent"
        required="true"/>
    <property name="componentType" //Commentaire : Nom correctement défini
    utilisé pour caractériser toutes les instances de ce composant
        path="sourceComponentId/componentType"
        required="true"
        defaultValue="ADAPTER"/>
    <property name="situation" //Commentaire : Définit le type de
    situation ayant entraîné le signalement de l'événement
        path="situation"
        required="true"/>
    <property name="categoryName" //Commentaire : Indique le type
    de situation pour l'événement
        path="situation/categoryName"
        required="true"
        permittedValue="CreateSituation"
        permittedValue="DestroySituation"
        permittedValue="OtherSituation" />
    <property name="situationType" //Commentaire : Indique le type
    de situation et la disposition de l'événement
        path="situation/situationType"
        required="true"
    <property name="reasoningScope" //Commentaire : Indique
    l'impact de l'événement
        path="situation/situationType/reasoningScope"
        required="true"
        permittedValue="INTERNAL"
        permittedValue="EXTERNAL"/>
    <property name="successDisposition" //Commentaire : Indique la
    réussite de l'événement
        path="situation/situationType/successDisposition"
        required="true"
        permittedValue="SUCCESSFUL"
        permittedValue="UNSUCCESSFUL" />
    <extendedDataElements name="Employee" //Commentaire : Nom de l'objet métier
    lui-même
        type="noValue"
        <children name="EmployeeId"
            type="string"/> //Commentaire : Le type est l'une des
    valeurs autorisées dans la documentation Common Event Infrastructure
        <children name="EmployeeAddress"
            type="noValue"/>
            <children name="EmployeeId"
                type="string"/>
            -
            -
            -
    </extendedDataElements
</eventDefinition>

```

Informations légales

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays.

Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire d'échange 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

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

*IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan*

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, EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. 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, à tout moment et sans préavis, modifier 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 Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.*

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 dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins

illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes en langage source, destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Informations sur les interfaces de programmation

Les informations sur les interfaces de programmation ont pour objectif de vous aider à créer des logiciels d'application à l'aide de ce programme.

Les interfaces de programmation génériques vous permettent de créer des logiciels d'application qui obtiennent les services des outils de ce programme.

Cependant, ces informations peuvent également contenir des informations sur le diagnostic, la modification et le réglage. Ces informations vous permettent d'exécuter le débogage de votre logiciel d'application.

Avertissement : N'utilisez pas ces informations sur le diagnostic, la modification et le réglage comme interface de programmation car elles sont susceptibles de changer.

Marques et marques de service

Les termes qui suivent sont des marques d'International Business Machines Corporation aux Etats-Unis et/ou dans certains autres pays :

i5/OS
IBM
le logo IBM
AIX
AIX 5L
CICS
CrossWorlds
DB2
DB2 Universal Database
Domino
HelpNow
IMS
Informix
iSeries
Lotus

Lotus Notes
MQIntegrator
MQSeries
MVS
Notes
OS/400
Passport Advantage
pSeries
Redbooks
SupportPac
WebSphere
z/OS

Java et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium et Pentium sont des marques ou marques déposées de Intel Corporation ou de ses filiales, aux Etats-Unis et dans d'autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Ce produit inclut un logiciel développé par Eclipse Project (<http://www.eclipse.org/>).



WebSphere Business Integration Adapters, Version 6.0



WebSphere Business Integration Adapter Framework V2.6.0

Index

A

- adaptateur JText
 - vérification de l'installation 17
- adaptateur pour e-mail
 - architecture 2
 - arrêt 26
 - compatibilité du courtier 13
 - composants 2
 - création de plusieurs instances 23
 - démarrage 24
 - environnement 13
 - exemple de scénario 8
 - exemples de fichiers 29
 - fonctionnalités 1
 - Généralités 1
 - initialisation 8
 - installation 17
 - vérification 17
 - installation, adaptateur pour e-mail
 - configuration 13
 - méta-objets
 - structure 40
 - planification de l'implémentation 27
 - Généralités 27
 - platesformes 14
 - tâches de configuration 18
 - traitement d'objets métier 16
 - utilisation des méta-objets 40
- adresse e-mail
 - indication 45
- appels Application Response
 - Measurement, prise en charge 93
- Application Response Measurement 11
- architecture 2
- architecture d'adaptateur 13
- archivage d'événement 7

C

- catalogue d'événements, pour Common Event Infrastructure 96
- Common Event Infrastructure 11
 - catalogue d'événements 96
 - métadonnées 96
- courtiers d'intégration 13

D

- déclenchement d'événement 4
- définition de l'adaptateur
 - configuration 19
- détection d'événement 4
- données dépendantes des paramètres régionaux
 - caractéristiques 10
 - priorités 10
 - traitement 10

E

- emplacement d'interrogation 5
- états des événements 5
- événements échoués 6
- extraction des événements 6

G

- gestionnaire de données
 - méta-objet 39

I

- IBM Tivoli Monitoring for Transaction Performance 11, 93
- installation
 - vérification
 - Unix 18
 - Windows 17
- Installation
 - vérification 17

J

- JavaMail
 - protocoles
 - IMAP 3
 - SMTP 3

M

- mécanisme de récupération 7
- messages d'erreur 1
- méta-objets
 - attributs de premier niveau 40
 - attributs enfants 40
 - ArchiveFolder 41
 - FailFolder 41
 - InDoubtEvents 42
 - Password 41
 - PollFolder 41
 - PollHostName 40
 - Reconnect 42
 - UserName 41
 - exemple 42

N

- notification d'erreur
 - utilisation 16
- notification d'événement 4
 - utilisation 15

O

- objet métier pour e-mail
 - attributs
 - indication d'objets métier 34, 36

- objet métier pour e-mail (*suite*)

- attributs facultatifs 33, 34
 - PriorityLevel 33
- attributs obligatoires
 - AttachmentExtension 32
 - BusinessObjectMimeCharset 32
 - BusinessObjectMimeEncoding 32
 - FromAddress 32
 - MessageContent 31
 - MessageContentMimeCharset 32
 - MessageContentMimeEncoding 32
 - MessageHeaderMimeCharset 32
 - MessageHeaderMimeEncoding 33
 - MessageSubject 31
 - MimeType 32
 - RecipientName 31
 - ReplyTo 33
- exemple 34
- objet métier pour E-mail
 - attributs
 - spécification des fichiers 37
 - attributs obligatoires 31
- objets métier
 - attributs EmailNotification 43
 - BusinessObjectMimeCharset 44
 - BusinessObjectMimeEncoding 44
 - FromAddress 44
 - MessageAttachment 44
 - MessageContent 43
 - MessageContentMimeCharset 44
 - MessageContentMimeEncoding 44
 - MessageHeaderMimeCharset 44
 - MessageHeaderMimeEncoding 44
 - MessageSubject 43
 - RecipientName 43
 - ReplyTo 44
 - conformité des gestionnaires de données 38
 - développement 31
 - EmailNotification 43
 - propriétés des attributs 39
 - Application -Specific Information 39
 - Foreign Key 39
 - Key 39
 - Max Length 39
 - Required 39
 - Type 39
 - valeurs par défaut 39
 - structures 31
 - traitement 3, 38
 - traitement des instructions 38
 - utilisation 31

P

- propriétés spécifiques à l'adaptateur 20
 - ApplicationUserName 21
 - BusinessObjectMimeCharset 21
 - BusinessObjectMimeEncoding 21
 - DataHandlerConfigMO 22

- propriétés spécifiques à l'adaptateur
(*suite*)
 - DebugMode 22
 - DefaultVerb 22
 - MailsPerMailBox 22
 - MessageContentMimeCharset 22
 - MessageContentMimeEncoding 22
 - MessageHeaderMimeCharset 22
 - MessageHeaderMimeEncoding 22
 - PollConfigMO 22
 - PollQuantity 23
 - SMTP_MailPort 23
 - UseDefaults 23
- propriétés spécifiques à l'application
 - ApplicationPassword 21
- propriétés standard de configuration 19
- propriétés standard de configuration
pour les connecteurs 95

S

- serveurs de messagerie
 - configuration de l'interrogation 19
- statut d'événement 5
- surveillance, des transactions 11, 93
- surveillance des transactions 11, 93

T

- Tivoli Monitoring for Transaction
Performance 11, 93

U

- Unix
 - structure de fichiers installée 18

W

- Windows
 - structure de fichiers installée 17

IBM