

IBM WebSphere InterChange Server



Introduction technique à IBM WebSphere InterChange Server

V4.3.0

IBM WebSphere InterChange Server



Introduction technique à IBM WebSphere InterChange Server

V4.3.0

octobre 2004

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 2004. Tous droits réservés.

© **Copyright International Business Machines Corporation 1997, 2004. All rights reserved.**

Table des matières

À propos de ce manuel	v
À qui s'adresse ce document ?	v
Conditions préalables pour ce document	v
Documents associés	v
Conventions typographiques	v
Nouveautés de cette version	vii
Nouveautés de WebSphere InterChange Server 4.3	vii
Nouveautés de WebSphere InterChange Server 4.2.2	vii
Nouveautés de WebSphere InterChange Server 4.2.1	vii
Chapitre 1. Présentation d'IBM WebSphere InterChange Server	1
InterChange Server et le système d'intégration de processus d'entreprise WebSphere	1
Modèle InterChange Server	2
Boîte à outils de WebSphere Business InterChange Server	3
Collaborations, objets métier et connectivité	3
Modèle de solutions d'implémentation.	4
Déploiement de plusieurs serveurs	6
Connectivité sur Internet	6
Flux des données dans une implémentation InterChange Server	9
Interactions publication-abonnement	10
Les requêtes d'accès	10
Interactions requête/réponse	11
Exemples de flux de données	13
Connecteurs	16
Communication du connecteur avec les applications	17
Liaisons entre les éléments	17
Liaison d'un déclencheur	17
Liaison des destinations	18
Objets métier	19
Rôles d'un objet métier	19
La structure d'un objet métier	20
Objets métier propres à une application et génériques.	22
Le mappage des données.	23
InterChange Server	25
Service de gestion des événements	25
Contrôleur de connecteur.	26
Référentiel.	26
Service de connectivité de la base de données	26
Pools de connexion à des bases de données.	27
Haute disponibilité	27
Service de transaction	27
Fonction de reprise sur incident	29
Infrastructure de transport des communications	30
Distribution sur un réseau	30
La distribution via Internet	32
La sécurisation d'InterChange Server	33
Le chiffrement	33
Confidentialité complète	34

Le contrôle d'accès basé sur les rôles	37
Récapitulatif	41

Chapitre 2. Outils à utiliser avec InterChange Server	43
Boîte à outils de WebSphere Business Integration	43
System Manager.	44
Configuration des composants	45
Modes de System Manager et d'InterChange Server	45
Outils de développement.	45
Outils d'administration	46

Chapitre 3. Collaborations	47
Objets et les modèles de collaborations	47
Traitement de la collaboration	48
Gestion des interventions sur appel et les processus métier longue durée	49
Collaborations et les traitements simultanés	49
Groupes de collaborations	50
Ports	50
Intervention sur appel dynamique.	52
Scénarios	53
Logique du processus métier	54
Interactions avec les connecteurs et les applications	56
Initialisation de la collaboration	58
Récapitulatif	58

Chapitre 4. Objets métier	59
Définitions des objets métier et les objets métier	59
Composants d'une définition d'objet métier	60
Attributs	60
Instructions	62
Présentation détaillée des objets métier propres à une application	62
Organisation des attributs	63
Informations propres à l'application	63
Les options de modification	64
Récapitulatif	65

Chapitre 5. Connecteurs	67
Initialisation du connecteur	67
Notification d'événement.	69
Configuration du mécanisme de notification d'événements de l'application	70
Détection d'un événement	73
Traitement d'un événement	73
Traitement des requêtes	75
Traitement à base d'instructions	76
informations propres à l'application basées sur des instructions	77
Capacités de traitement simultané.	78
Construction et déconstruction des objets métier	78

Métadonnées des objets métier et les actions des connecteurs	79
Avantages des agents du connecteur basés sur les métadonnées	80
Exemple de construction d'objet métier	80
Configuration du connecteur	81
Propriétés du connecteur	82
Mappes associées	82
Développement du connecteur	82
Récapitulatif	83

Chapitre 6. Mappage de données 85

Comment le système InterChange Server utilise le mappage	85
Composants et outils des mappes	87
Transformations du mappage	88
Transformations simples	88
Transformations des relations	89
Configuration des connecteurs avec des mappes	90
Récapitulatif	90

Chapitre 7. Collaborations

transactionnelles 91

Modèle de transaction	91
Qu'est-ce que la collaboration transactionnelle ?	92
Scénarios transactionnels	92
Sous-transactions	93
Correction et l'invalidation	95
Isolement des données.	97
Niveaux de transaction	98
Aucun	99
Niveau minimal	99
Niveau maximal	99
Strict	100
Reprise sur incident	100
Collaborations transactionnelles et les processus métier longue durée	101
Récapitulatif	101

Chapitre 8. Prise en charge des comportements spécifiques aux langues 103

Prise en charge de l'environnement local dans les produits WebSphere Business Integration	103
Installation d'un environnement local	104
Traitement des données dépendantes de l'environnement local.	104
Remarques relatives à la conception	105
Codage des données de contenu	105
Codage des métadonnées de configuration	106
Codage des données de fichier journal et de trace	106
Prise en charge des données bidirectionnelles.	106
Caractéristiques des langues bidirectionnelles	107
Attributs et transformations de la présentation	110
Présentation du texte	110
Attributs de présentation	111
Transformations de présentation	112
Prise en charge des scripts bidirectionnels dans les produits WebSphere Business Integration	112
Prise en charge des scripts bidirectionnels	113
Prise en charge des scripts bidirectionnels par Adapter Framework	113
Prise en charge des collaborations pour les scripts bidirectionnels.	114
Prise en charge des scripts bidirectionnels par les mappes	115
Gestion du texte bidirectionnel	115
Migration de données	115
Chaînes bidirectionnelles spéciales	116
API BiDi	116
limitations conceptuelles.	116
Récapitulatif	118

Remarques 119

Informations sur l'interface de programmation	120
Marques de commerce et marques de service.	121

Index 123

À propos de ce manuel

IBM WebSphere InterChange Server et ses boîtes à outils associées sont utilisés avec les adaptateurs d'intégration de processus d'entreprise d'IBM WebSphere pour apporter l'intégration de processus et la connectivité nécessaires aux principales technologies de commerce électronique et aux applications des entreprises.

Ce document présente l'infrastructure et les principaux éléments d'un système utilisant InterChange Server comme courtier d'intégration du système d'intégration de processus d'entreprise WebSphere.

À qui s'adresse ce document ?

Ce document est destiné aux consultants IBM et aux clients.

Conditions préalables pour ce document

Ceci est un document d'introduction, à lire préalablement à l'utilisation de toutes les autres documentations InterChange Server.

Documents associés

L'ensemble des documentations disponibles pour ce produit décrit les caractéristiques et les composants communs à toutes les installations WebSphere InterChange Server, et comprend des documents de référence sur des composants spécifiques.

Vous pouvez consulter la documentation à partir des sites suivants :

- Pour la documentation InterChange Server :
<http://www.ibm.com/websphere/integration/wicserver/infocenter>
- Pour la documentation sur les collaborations :
<http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>
- Pour la documentation sur WebSphere Business Integration Adapters :
<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

Ces sites vous guideront de façon simple pour télécharger, installer et consulter la documentation.

Conventions typographiques

Ce document utilise les conventions suivantes :

<i>police courrier</i>	Indique une valeur littérale, telle qu'un nom de commande, un nom de fichier, des informations que vous saisissez ou bien des informations que le système affiche à l'écran.
gras	Indique un nouveau terme la première fois qu'il apparaît.
<i>italique,italique</i>	Indique un nom de variable ou une référence croisée.

<i>texte en bleu</i>	Le texte en bleu, que vous ne pouvez distinguer que lorsque vous consultez le manuel en ligne, indique le lien hypertexte d'une référence croisée. Cliquez sur le texte en bleu pour afficher l'objet de la référence.
{ }	Dans une ligne de syntaxe, les accolades entourent un ensemble d'options dont il ne faut choisir qu'une seule.
[]	Dans une ligne de syntaxe, les crochets entourent un paramètre facultatif.
...	Dans une ligne de syntaxe, les points de suspension indiquent la répétition d'un paramètre précédent. Par exemple, <code>option[,...]</code> signifie que vous pouvez saisir de multiples options séparées par des virgules.
< >	Dans la convention de dénomination, les crochets entourent les éléments indépendants d'un nom pour les distinguer les uns des autres, comme dans <code><server_name><connector_name>tmp.log</code> .
/, \	Dans ce document, les barre oblique inverse (\) sont utilisées dans la composition conventionnelle des chemins d'accès aux répertoires. Pour les installations UNIX, remplacez les barres obliques (/) par des barres obliques inverses. Tous les noms de chemins d'accès aux répertoires du produit IBM WebSphere dans cette documentation font référence aux répertoires dans lesquels le produit IBM WebSphere est installé sur votre système.
<i>%texte% and \$texte</i>	Un texte entouré par des signes pour-cent (%) indique la valeur du texte de la variable système Windows ou de la variable utilisateur. La variable équivalente dans un environnement UNIX est <i>\$texte</i> , indiquant la valeur de la variable <i>texte</i> de l'environnement UNIX.
<i>ProductDir</i>	Représente le répertoire dans lequel le produit est installé. Pour l'environnement IBM WebSphere InterChange Server, le répertoire par défaut est "IBM\WebSphereICS". Pour l'environnement IBM WebSphere Business Integration Adapters, le répertoire par défaut est "WebSphereAdapters".

Nouveautés de cette version

Ce chapitre décrit les nouvelles fonctions du système WebSphere business integration qui sont développées dans ce document.

Nouveautés de WebSphere InterChange Server 4.3

La version 4.3 d'IBM WebSphere InterChange Server offre les fonctions suivantes :

- La confidentialité complète, qui sécurise les messages dès leur émission sur le noeud d'origine, à travers tous les noeuds intermédiaires et jusqu'au noeud de destination. La confidentialité complète fournit à InterChange Server une sécurité contre la macération, asymétrique et dynamique.
- Le contrôle des accès basé sur les rôles authentifie les utilisateurs qui veulent accéder à InterChange Server, puis limite l'accès des utilisateurs à certains composants individuels, en fonction du rôle qui a été attribué à l'utilisateur.
- Les interventions sur appel dynamiques permettent à la collaboration d'effectuer une intervention sur appel vers une destination qui n'avait pas été prédéfinie et liée explicitement pendant le développement de la collaboration.
- Prise en charge d'IBM Tivoli License Manager (ITLM)
- Prise en charge des comportements spécifiques aux langues

Nouveautés de WebSphere InterChange Server 4.2.2

IBM WebSphere InterChange Server 4.2.2 offre les fonctions suivantes :

- InterChange Server (ICS) et les autres composants d'ICS utilisent désormais le répartiteur de requêtes d'objets IBM Java Object Request Broker (ORB) au lieu du répartiteur de requêtes d'objets tiers VisiBroker.

Nouveautés de WebSphere InterChange Server 4.2.1

Ce guide fournit les descriptions fonctionnelles des nouvelles fonctions de System Manager pour la version 4.2.1 d'IBM WebSphere InterChange Server 4.2.1.

Chapitre 1. Présentation d'IBM WebSphere InterChange Server

Ce document fournit une introduction fonctionnelle à l'utilisation d'IBM WebSphere InterChange Server en tant que courtier d'intégration dans un système d'intégration de processus d'entreprise IBM WebSphere. Dans un système d'intégration de processus d'entreprise, InterChange Server fournit une infrastructure distribuée permettant de résoudre les incidents inter-application, y compris la possibilité de :

- Déplacer des informations métier entre plusieurs sources pour réaliser des échanges métier via Internet, et
- Traiter et acheminer des informations métier entre des applications hétérogènes dans l'environnement de l'entreprise

Ce chapitre fournit une présentation générale de l'architecture, des composants et du flux du procédé d'un système d'intégration qui utilise InterChange Server en tant que courtier d'intégration. Les sections suivantes seront abordées :

- «InterChange Server et le système d'intégration de processus d'entreprise WebSphere»
- «Modèle InterChange Server» à la page 2
- «Boîte à outils de WebSphere Business InterChange Server» à la page 3
- «Collaborations, objets métier et connectivité» à la page 3
- «Flux des données dans une implémentation InterChange Server» à la page 9
- «Connecteurs» à la page 16
- «Liaisons entre les éléments» à la page 17
- «Objets métier» à la page 19
- «Le mappage des données» à la page 23
- «InterChange Server» à la page 25
- «Infrastructure de transport des communications» à la page 30
- «La sécurisation d'InterChange Server» à la page 33
- «Récapitulatif» à la page 41

Remarque : Tout au long de ce manuel des illustrations sont données à titre d'exemple (uniquement), afin de présenter la structure et les concepts. Elles ne documentent pas nécessairement des composants spécifiques réels.

InterChange Server et le système d'intégration de processus d'entreprise WebSphere

IBM WebSphere InterChange Server et ses boîtes à outils associées sont utilisés avec les adaptateurs d'intégration de processus d'entreprise IBM WebSphere dans un **système d'intégration de processus d'entreprise WebSphere**. Au niveau fonctionnel le plus élevé, un système d'intégration de processus d'entreprise WebSphere est composé d'un courtier d'intégration et d'un ensemble d'adaptateurs qui permettent à des applications métier hétérogènes d'échanger des données via un transfert coordonné des informations, sous la forme d'objets métier.

InterChange Server n'est pas le seul courtier d'intégration pouvant être utilisé comme élément central d'un système d'intégration de processus d'entreprise WebSphere. D'autres systèmes utilisent WebSphere MQ Integrator Broker, WebSphere Business Integration Message Broker, ou WebSphere Application Server. Tous ces courtiers d'intégration peuvent utiliser les adaptateurs d'intégration de processus d'entreprise de WebSphere, qui utilisent les connecteurs pour la connectivité des applications et les objets métier pour contenir les données de gestion échangées entre les applications. Cependant, il existe certaines différences entre les systèmes d'intégration utilisant les deux différents courtiers, et il est important de choisir le courtier qui correspond le mieux à vos besoins du point de vue fonctionnel.

Ce guide met principalement l'accent sur les concepts du système d'intégration qui sont mis en oeuvre avec InterChange Server utilisé en tant que courtier d'intégration. Pour les informations concernant l'implémentation d'un système d'intégration utilisant l'un des courtiers de messages WebSphere (WebSphere MQ Integrator, WebSphere MQ Integrator Broker, ou WebSphere Business Integration Message Broker) en tant que courtier d'intégration, consultez le guide *Implementing Adapters with WebSphere Message Brokers* (Implémentation des adaptateurs avec les courtiers de messages WebSphere). Pour les informations concernant l'implémentation d'un système d'intégration utilisant WebSphere Application Server en tant que courtier d'intégration, consultez le guide *Implementing Adapters with WebSphere Application Server* (Implémentation des adaptateurs avec WebSphere Application Server). Ces deux guides d'implémentation font partie du jeu de documentations sur les adaptateurs d'intégration de processus d'entreprise WebSphere.

Modèle InterChange Server

Un système d'intégration de processus d'entreprise WebSphere implémenté avec InterChange Server utilise des composants modulaires ainsi qu'une logique métier indépendante des applications. L'approche d'InterChange Server est basée sur une infrastructure distribuée selon le schéma de la roue à rayons (hub-and-spoke). Cela vous permet d'exécuter les distributions d'ordres des processus métier, le traitement des retours, distribués via Internet, via des applications sur des réseaux locaux, par exemple, ou via les deux. Le système est réparti et souple, avec des composants réutilisables et des fonctionnalités de personnalisation qui permettent de répondre aux besoins spécifiques des sites et des applications.

Une implémentation d'InterChange Server utilise les connecteurs fournis par le produit WebSphere Business Integration Adapters, ainsi que plusieurs types de composants modulaires et personnalisables, englobant les collaborations, les objets métier et les gestionnaires de données. Les chapitres qui suivent dans ce guide décrivent l'utilisation de ces composants dans un système d'intégration de processus d'entreprise WebSphere avec InterChange Server en tant que courtier.

IBM WebSphere InterChange Server fournit les droits permettant son utilisation avec IBM Tivoli License Manager. IBM WebSphere InterChange Server assure le support pour les fonctions de gestion de licences et d'inventaire pour les composants serveurs et le Toolset. Voir *Access Development Guide for Enterprise Java Beans* et *Access Development Guide J2EE Connector Architecture* pour plus d'informations.

Boîte à outils de WebSphere Business InterChange Server

Avec InterChange Server, vous disposez d'une boîte à outils WebSphere Business Integration qui vous permet :

- De développer et de personnaliser des composants
- D'organiser vos composants en projets pour un déploiement en étapes successives de développement et de production
- De gérer un système d'intégration InterChange Server déployé

Ces opérations sont décrites dans le Chapitre 2, «Outils à utiliser avec InterChange Server», à la page 43.

Collaborations, objets métier et connectivité

Remarque : Dans ce document, "application" se réfère à un progiciel d'entreprise qui est en cours d'intégration ; il ne s'agit *pas* d'un composant d'IBM WebSphere InterChange Server.

Le système IBM WebSphere InterChange Server utilise une infrastructure centrale (InterChange Server) ainsi que des éléments modulaires, dans une conception selon le schéma de la roue à rayons (hub-and-spoke), de la façon suivante :

- La logique des processus métier réside dans les **collaborations** WebSphere InterChange Server sur le concentrateur.

Les collaborations InterChange Server sont des modules logiciels qui contiennent une logique décrivant un processus métier distribué. Il existe différentes collaborations pour des processus métier fondamentaux différents, tels qu'une collaboration GestionContacts, ou une collaboration MouvementInventaire. Les collaborations coordonnent les fonctionnalités des processus métier pour des applications hétérogènes et permettent les échanges de fichiers entre-elles. Les collaborations constituent le concentrateur : à travers elles, les données sont échangées sous la forme d'objets métier.

- Les données sont échangées entre le concentrateur et les applications distantes sous la forme d'**objets métier**.

Les objets métier sont des messages utilisés par le système IBM WebSphere InterChange Server pour échanger des données. Les **gestionnaires de données** sont utilisés pour transformer les données applicatives en série en objets métier, et les **mappes** sont utilisées entre un objet métier qui est structuré pour le modèle de données d'une application spécifique et un objet métier qui est structuré de façon générique pour être utilisé par les collaborations sur le concentrateur.

- Application et technologie Les **connecteurs**, qui sont disponibles dans le produit IBM WebSphere Integration Adapters, fournissent la connectivité aux applications (ou aux serveurs web et autres entités par programmation) distantes.

Les connecteurs peuvent être configurés pour interagir soit dans le cadre d'un réseau, soit via Internet et au-delà des pare-feu.

Chaque connecteur est composé de deux parties : le **contrôleur de connecteur** et l'**agent du connecteur**. Le contrôleur de connecteur interagit directement avec les objets de collaboration WebSphere InterChange Server et réside sur un serveur qui a implémenté le système IBM WebSphere InterChange Server (le concentrateur dont la relation s'effectue selon le schéma de la roue à rayons (hub-and-spoke). L'agent du connecteur interagit directement avec une application et peut résider avec cette application sur n'importe quel serveur. Un

agent distant peut être utilisé pour implémenter une communication entre un contrôleur de connecteur sur un concentrateur et un agent qui réside sur un autre site via Internet.

Certains connecteurs sont conçus pour interagir avec des applications spécifiques. Les connecteurs d'applications, tel le connecteur Clarify, sont des intermédiaires entre les collaborations et les applications. Ces connecteurs transforment les données provenant de l'application en objets métier qui peuvent être pris en charge par les collaborations, et transformer les objets métier provenant des collaborations en données qui peuvent être reçues par l'application spécifique.

D'autres connecteurs sont conçus pour des interactions conformes à des normes technologiques spécifiques. (Par exemple, le connecteur XML peut être utilisé pour envoyer des données provenant de collaborations InterChange Server vers un serveur Web, même si ce serveur Web réside au-delà d'un pare-feu sur un réseau qui n'exécute pas l'agent du connecteur ou un autre logiciel IBM WebSphere.)

- L'**interface Server** permet aux sites de destination distants qui n'implémentent pas WebSphere InterChange Server d'utiliser des **clients d'accès**, qui effectuent des appels via Internet vers un site concentrateur qui ne dispose pas d'InterChange Server.

L'interface d'accès au serveur fait partie d'InterChange Server. Il s'agit d'une interface de programme d'application compatible CORBA qui accepte les transferts de données synchrones aussi bien en provenance d'un réseau local que de sources externes. Les données sont alors transformées en objets métier qui peuvent être pris en charge par une collaboration. L'interface d'accès au serveur permet de recevoir des appels d'entités externes (par exemple en provenance de navigateurs Web sur les sites distants des clients) qui ne passent pas à travers des agents du connecteur, mais à travers des servlets Web dans l'interface d'accès au serveur.

L'interface d'accès au serveur et les connecteurs font tous deux appel aux gestionnaires de données. Dans l'environnement InterChange Server, de nouveaux gestionnaires de données peuvent être créés à partir d'un groupe modulaire de classes de base appelé **Data Handler Framework** (structure du gestionnaire de données). La solution InterChange Server comprend également un gestionnaire de protocole intégré : **Protocol Handler Framework**. Ces structures facilitent la personnalisation des solutions et ajoute de la connectivité pour les formats de données et les protocoles supplémentaires que vous serez amené à gérer ultérieurement.

Modèle de solutions d'implémentation

Une solution InterChange Server standard comprend une ou plusieurs collaborations ainsi qu'un ensemble d'objets métier qui représentent des informations métier qui concernent l'entreprise. Les collaborations et les objets métier s'utilisent avec des connecteurs, l'interface d'accès au serveur, ou bien les deux.

Un connecteur peut interagir avec une ou plusieurs collaborations, exécutant ainsi plusieurs processus métier. Et chaque collaboration peut interagir avec un nombre illimité de connecteurs, impliquant ainsi un nombre illimité d'applications.

Par exemple, pour la mise à jour automatique d'une application de planification des ressources d'entreprise (ERP), lorsqu'une information client est modifiée dans une application de gestion des interactions clients (CIM), la solution InterChange Server devrait être constituée de la collaboration ClientSync, de connecteurs pour

l'application CIM et l'application ERP, ainsi que de définitions d'objets métier qui représentent les informations client. La figure 1 illustre cette solution.

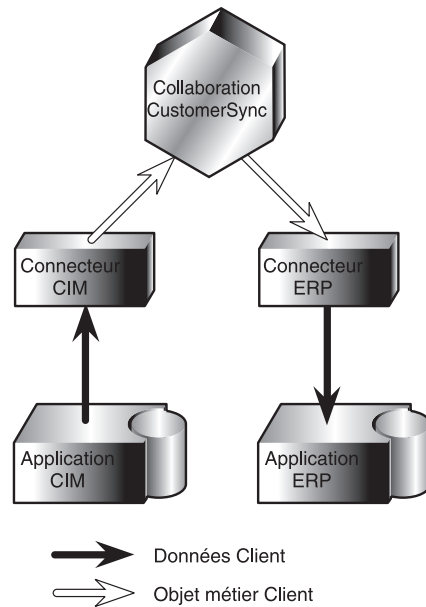


Figure 1. Solution de données clients CIM vers ERP

La solution de la figure 1-1 pourrait être implémentée soit localement sur un réseau, soit via Internet.

Dans la figure 2, sur un site qui n'a implémenté ni le système IBM WebSphere InterChange Server ni un connecteur, un représentant du client pourrait souhaiter utiliser un navigateur Web pour obtenir le statut d'un bon de commande via Internet à partir d'une application ERP (SAP dans l'exemple) qui se trouve sur un site qui utilise le système WebSphere InterChange Server. Pour permettre ceci, la solution InterChange Server utilise l'interface d'accès au serveur, avec une collaboration (fictive dans cet exemple) pour la logique métier du bon de commande, un connecteur SAP et des objets métier qui représentent les informations sur le statut du bon de commande.

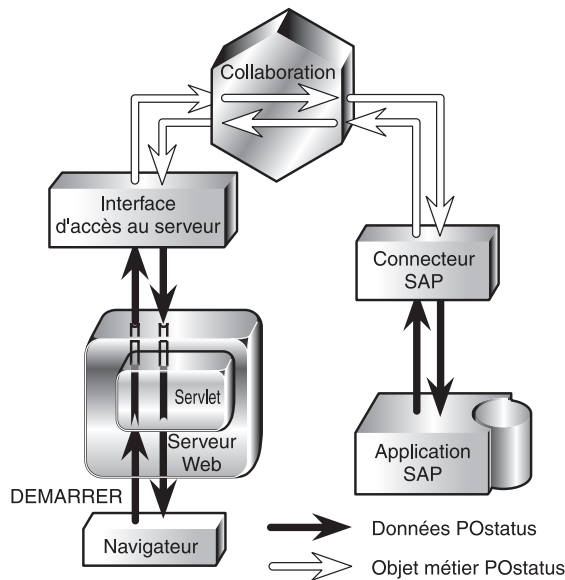


Figure 2. Exécution d'un appel via l'interface d'accès au serveur

Déploiement de plusieurs serveurs

Une application unique peut interagir avec plusieurs instances d'InterChange Server, dans un environnement à concentrateur en étoile. Plusieurs serveurs InterChange Server peuvent être déployés pour travailler ensemble. Les agents du connecteur peuvent être configuré pour partitionner et acheminer les informations d'une application vers différents serveurs. Ceci permet d'équilibrer la charge de traitement d'une application entre plusieurs instances d'InterChange Server. De plus, les serveurs InterChange Server peuvent être configurés pour interagir les uns avec les autres via les connecteurs et l'interface d'accès au serveur.

Connectivité sur Internet

Les connecteurs peuvent permettre l'échange de données via Internet de différentes façons. Ces possibilités sont :

- Les agents du connecteur de l'application implémentés à distance

Cette technique (appelée technologie de l'agent à distance) utilise JMS, avec soit SSL en WebSphere MQ natif soit HTTP/HTTPS dans WebSphere MQ internet pass-thru (MQIPT), pour implémenter un connecteur sur Internet. Un site unique qui a implémenté le système IBM WebSphere InterChange Server se comporte comme un concentrateur et gère des échanges sur Internet avec des sites distants qui ont implémenté un agent du connecteur éloigné. agent.

- Connecteurs de technologies utilisés avec l'interface d'accès au serveur

Dans cette approche, l'interface d'accès au serveur est utilisée pour effectuer des appels synchrones dans IBM WebSphere InterChange Server ; les connecteurs qui utilisent les normes technologiques d'Internet sont utilisés pour envoyer des données à partir d'IBM WebSphere InterChange Server :

- Les appels provenant de navigateurs externes (ou d'autres sources Internet) peuvent être reçus par un servlet Web qui envoie l'appel via l'interface d'accès au serveur pour déclencher des processus métier dans des collaborations.

- Pour le compte des collaborations, les connecteurs spécifiquement dédiés à des normes technologiques Internet (tel le XML) engagent une interaction requête/réponse avec des destinations externes qui comprennent le format de données spécifique.

L'interface d'accès au serveur réside sur un site concentrateur d'InterChange Server (ICS). Lorsque l'interface d'accès au serveur reçoit un appel, elle envoie les données vers un gestionnaire pour ce format de données spécifique (tel le gestionnaire de données XML). Le gestionnaire de données transforme les données en objet métier générique. L'interface d'accès au serveur envoie alors l'objet métier à la collaboration. La collaboration exécute ses processus sur les objets métier et répond, et cette réponse est retransformée dans le format de données spécifique qui était utilisé au début du processus.

Pour accepter des appels provenant de processus externes et envoyer les appels sous la forme d'objets métier vers une collaboration, l'interface d'accès au serveur nécessite les éléments suivants :

- Un serveur Web qui utilise un servlet ou tout autre processus pour envoyer des données à l'interface d'accès au serveur. Les données doivent être du type MIME pour lequel les gestionnaires de données ont été configurés dans InterChange Server. Le servlet utilise l'interface d'accès au serveur pour appeler un gestionnaire de données qui va convertir les données dans un format d'objet métier, puis il utilise l'interface d'accès au serveur pour envoyer l'objet métier sous forme d'appel à la collaboration.
- Une collaboration qui a été configurée pour gérer des requêtes qui sont reçues sous forme d'appels via l'interface d'accès au serveur.

Un exemple de cette approche est illustré dans la figure 3.

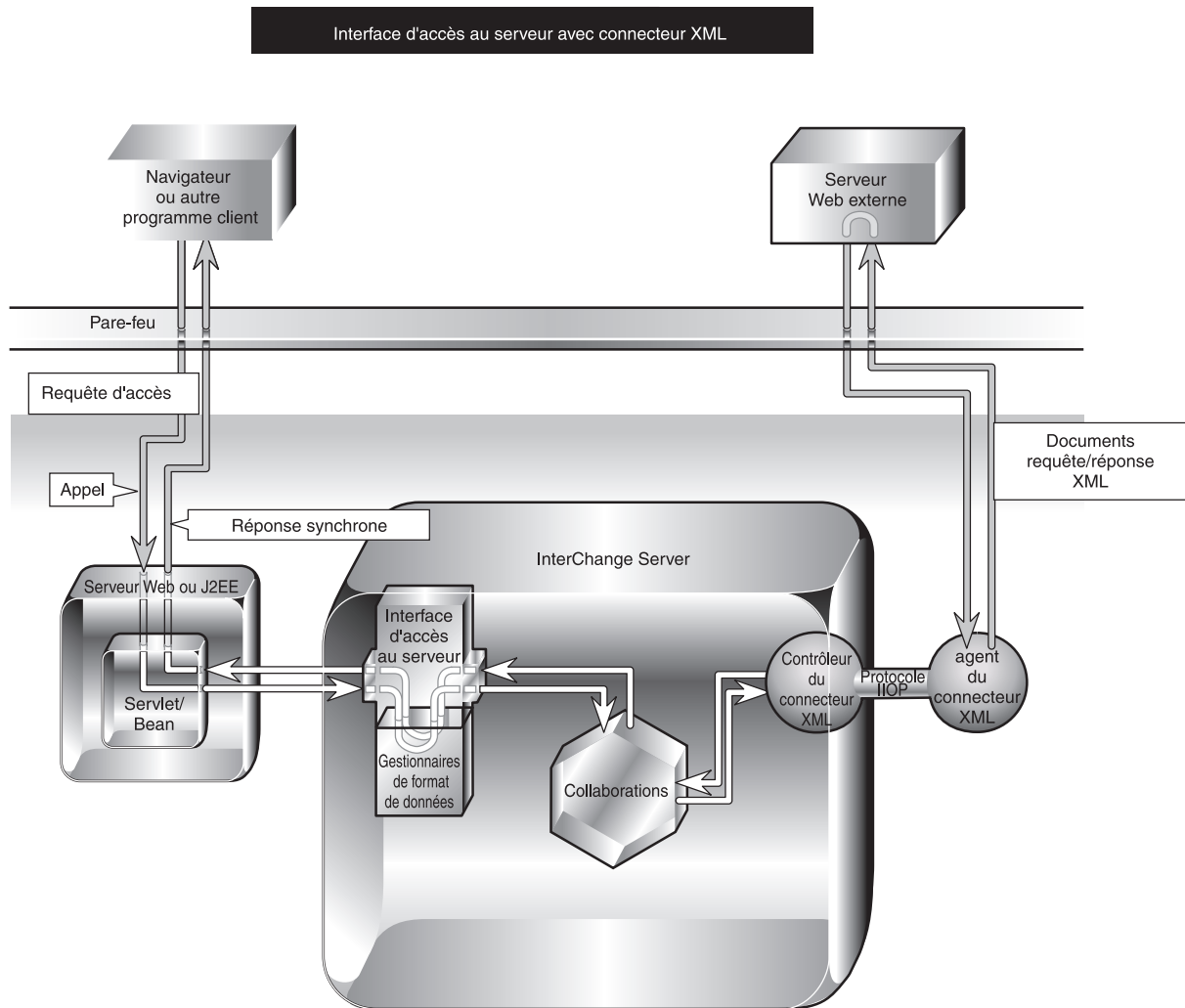


Figure 3. Échange de données via Internet à l'aide de l'interface d'accès au serveur

- Les connecteurs de technologies conçus pour les échanges Internet
Certains connecteurs peuvent permettre à la fois des interactions publication-abonnement et requête/réponse qui échangent les données via Internet, sans utiliser une configuration d'agent distant. Par exemple, le connecteur TPI permet l'échange de données en format XML, EDI ou binaire avec des partenaires d'échanges distants via Internet. D'une façon similaire, le connecteur Courriel échange des données via Internet à l'aide du protocole SMTP.

Un exemple de cette approche est illustré dans la figure 4.

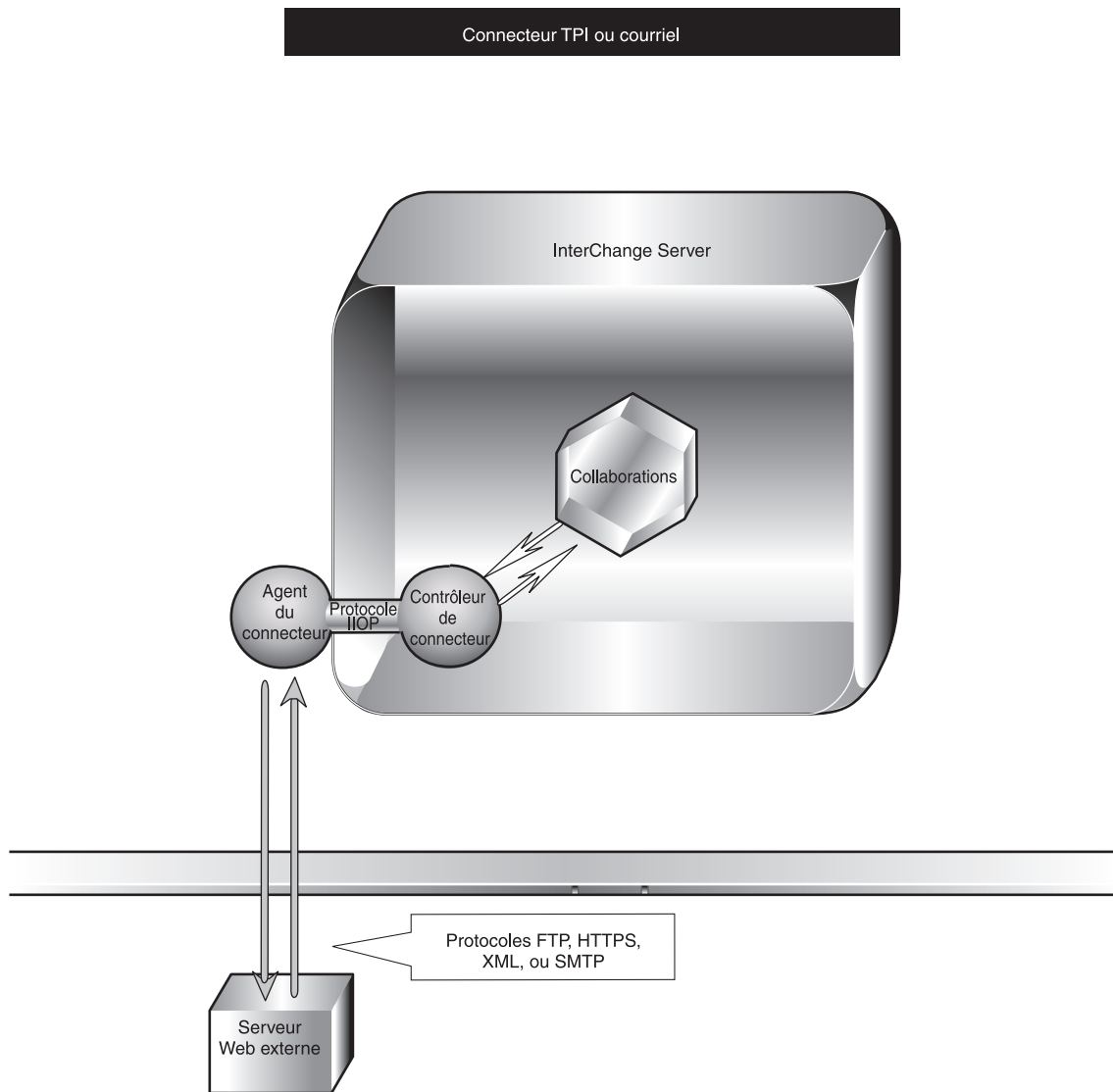


Figure 4. Échange de données via Internet à l'aide du connecteur TPI ou Courriel

Flux des données dans une implémentation InterChange Server

Dans une implémentation InterChange Server, un flux de données (le mouvement et le traitement des données d'une application ou entité à une autre) peut comprendre tous les éléments suivants :

- Des échanges asynchrones ou synchrones entre des applications hétérogènes sur un réseau local.

- Des échanges asynchrones ou synchrones entre des applications hétérogènes sur via Internet.
- Des échanges asynchrones ou synchrones entre des applications en réseau local et des serveurs Web externes.
- Des processus métier complets qui intègrent tous les éléments ci-dessus.

Un flux de données peut être initié par l'un des deux types d'interactions suivantes : des interactions publication-abonnement et des interactions d'interventions sur appel. Les deux types d'interactions sont à l'origine du déclenchement qui lance l'exécution du processus métier de la collaboration. La collaboration utilise alors un troisième type d'interaction (requête/réponse) pour terminer l'échange de données avec la destination souhaitée.

Interactions publication-abonnement

Les connecteurs et les collaborations utilisent une interaction **publication-abonnement** pour transférer les informations relatives aux événements d'applications vers IBM InterChange Server afin qu'elles y soient traitées.

Lors d'une interaction publication-abonnement, une collaboration démarre son processus métier lorsqu'un déclencheur lui demande de le faire par la réception d'un objet métier pour un type particulier d'**événement déclencheur**, (par exemple *Employé.Créer*, qui représente une opération de l'application). Le nom de l'objet métier (*Employé*) indique un type d'entité de gestion. L'instruction (*Créer*) indique l'opération qui s'est déroulée sur cette entité. Par conséquent, l'événement *Employé.Créer* rend compte de la création d'une entité employé.

L'interaction publication-abonnement permet à un événement déclencheur d'attendre une collaboration de la façon suivante :

- Une collaboration **s'abonne** à l'événement qui peut déclencher son exécution. Une collaboration s'abonne à un événement en effectuant une requête dans ce sens, puis en attendant l'événement. Une collaboration qui s'abonne à *Employé.Créer* démarre son traitement lorsque l'objet métier pour l'événement *Employé.Créer* est reçu.
- Un événement qui survient dans une application est alors détecté par le mécanisme de **notification d'événement** du connecteur de l'application. Le connecteur fournit l'événement à une ou plusieurs collaborations en le publiant : en le rendant disponible en tant qu'objet métier.

En fonction du connecteur, un événement peut être publié à la collaboration soit de façon asynchrone, soit de façon synchrone. De plus, si la fonctionnalité 'processus métier longue durée' de la collaboration est activée, une collaboration peut maintenir l'événement en attente, pour anticiper des événements entrants satisfaisant à des critères de correspondance prédéfinis.

Les requêtes d'accès

Une collaboration peut être conçue pour être déclenchée par des appels directs qui sont envoyés par un **client d'accès**, reçus par l'interface d'accès au serveur et envoyés à la collaboration en tant qu'objets métier. Dans une implémentation InterChange Server, les appels envoyés aux collaborations, par l'intermédiaire de l'interface d'accès au serveur, sont appelés **requêtes d'accès**. Les requêtes d'accès peuvent provenir de sources externes ou de sources qui sont configurées dans l'implémentation même d'InterChange Server.

Les interactions de requêtes d'accès sont utiles lorsqu'il est nécessaire d'effectuer une communication synchrone, comme par exemple lorsqu'un représentant du client utilise un navigateur Web pour demander des informations sur l'état des stocks via Internet.

Interactions requête/réponse

Une collaboration commence à traiter des données lorsqu'elle en reçoit l'ordre, qui est déclenché par l'arrivée d'un objet métier de déclenchement. L'objet métier de déclenchement peut résulter d'une requête d'accès ou d'une notification d'événement. Une fois la collaboration déclenchée, elle peut effectuer des requêtes auprès des connecteurs auxquels elle a été liée, et recevoir ainsi des réponses.

La collaboration effectue ses requêtes, appelées **requêtes d'intervention sur appel** sous la forme d'objets métier génériques. Les connecteurs transforment les objets métier génériques en entités de données qui sont interprétées par l'application ou le format de données spécifiques pour lesquels le connecteur a été conçu.

Les réponses que les collaborations reçoivent des connecteurs (appelées **réponse à une intervention sur appel**) peuvent prendre la forme d'objets métier contenant des données de gestion (dans le cas de requêtes d'extraction) ou rapports d'état (succès ou échec).

La figure 5 montre une vue simplifiée d'un processus métier qui est initié par une interaction publication-abonnement et qui s'achève avec des interactions requête/réponse. (Dans ce schéma simplifié, les contrôleurs de connecteur et les agents du connecteur sont représentés comme des unités individuelles, et aucune distinction n'est réalisée entre les connecteurs des applications en réseau local et les connecteurs des applications qui se trouvent sur Internet au-delà d'un pare-feu). L'exemple présente un processus métier distribué qui génère automatiquement une facture lorsqu'un représentant du service clientèle termine son travail sur un cas. Dans cet exemple, une hypothétique collaboration de service de facturation utilise des connecteurs pour échanger des données de gestion avec trois applications différentes.

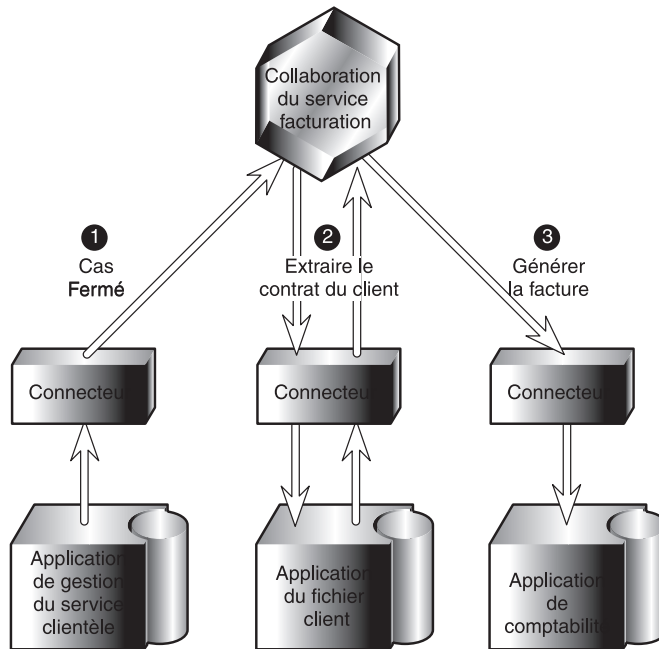


Figure 5. Une interaction publication-abonnement et requête/réponse

La figure 5 illustre la séquence suivante dans le processus métier :

1. Un représentant du service clientèle termine de travailler sur un cas. Le connecteur détecte la fermeture du cas en tant qu'événement dans l'application de gestion du service clientèle et extrait les données pertinentes du cas. Le connecteur publie alors l'événement, le rendant ainsi disponible pour une collaboration qui y est abonnée. Ces actions comprennent une interaction publication-abonnement.
2. Pour pouvoir calculer le montant de la facture, la collaboration a besoin de connaître les dispositions du contrat du client. La collaboration envoie une requête d'appel pour extraire les données nécessaires du connecteur de l'application qui gère les fichiers clients. Le connecteur répond à la requête. Ces actions incluent une interaction requête/réponse entre la collaboration et l'application et le connecteur qui gèrent les fichiers clients.
3. En utilisant à la fois les informations relatives au cas et au contrat du client, la collaboration produit les informations nécessaires pour générer la facture. Elle envoie la requête de création de facture au connecteur de l'application de comptabilité, qui envoie la requête à l'application elle-même, et répond à la collaboration par une notification de succès ou d'échec. Ces actions incluent une interaction requête/réponse entre la collaboration et l'application de comptabilité et son connecteur.

Un site peut ajuster le rapprochement avec lequel les collaborations et les connecteurs sont couplés. La collaboration pourrait, par exemple, fonctionner 24 heures sur 24, envoyant des requêtes à un connecteur qui communique avec son application uniquement entre minuit et 2 heures du matin. La collaboration peut être conçue et configurée pour envoyer des requêtes sans attendre de réponse et de tout simplement traiter les réponses lorsqu'elles arrivent.

D'un autre côté, une collaboration qui a été activée pour des processus métier longue durée peut préserver le contexte de flux d'une requête, et envoyer la requête avec une valeur de délai d'expiration, spécifiant la durée pendant laquelle une réponse peut permettre la reprise du flux de traitement enregistré.

Exemples de flux de données

La figure 6 présente une vue fonctionnelle des composants du système IBM WebSphere InterChange Server utilisés lors des interactions de requête d'accès standard, publication-abonnement, et requête/réponse, pour transférer des données entre des applications (ou d'autres entités tels des serveurs et navigateurs Web).

Les données se déplacent dans un flux à travers InterChange Server sur le concentrateur, puis sont échangées avec : les applications sur le réseau local ; les applications configurées avec des agents du connecteur au-delà d'un pare-feu Internet ; et les entités externes tels les serveurs et les navigateurs Web.

Le flux peut être initié par une requête d'accès ou par un événement ou tout type de processus interne ou externe. Une fois le flux initié, la collaboration utilise l'interaction requête/réponse pour terminer le processus métier soit avec des applications locales soit avec des application éloignée ou tout autre type d'entités.

Dans ce diagramme, un connecteur est en train de publier un événement d'application vers des collaborations, un autre connecteur s'engage dans une interaction requête/réponse entre une collaboration et une destination serveur Web. Notez bien que ceci n'illustre qu'une possibilité de configuration : la plupart des connecteurs peuvent être configurés pour à la fois publier des événements et répondre aux requêtes des collaborations.

La possibilité de configuration dans laquelle un connecteur interagit avec un serveur à technologie configurée localement, tel qu'un serveur TPI ou un serveur de courriel, pour échanger des données via Internet ne figure pas dans ce graphique.

De plus, notez bien qu'un agent du connecteur et ses applications connexes n'ont pas besoin de résider localement. En utilisant une technologie agent éloignée, les agents du connecteur qui se trouvent sur des sites distants peuvent à la fois publier des requêtes de collaboration et y répondre.

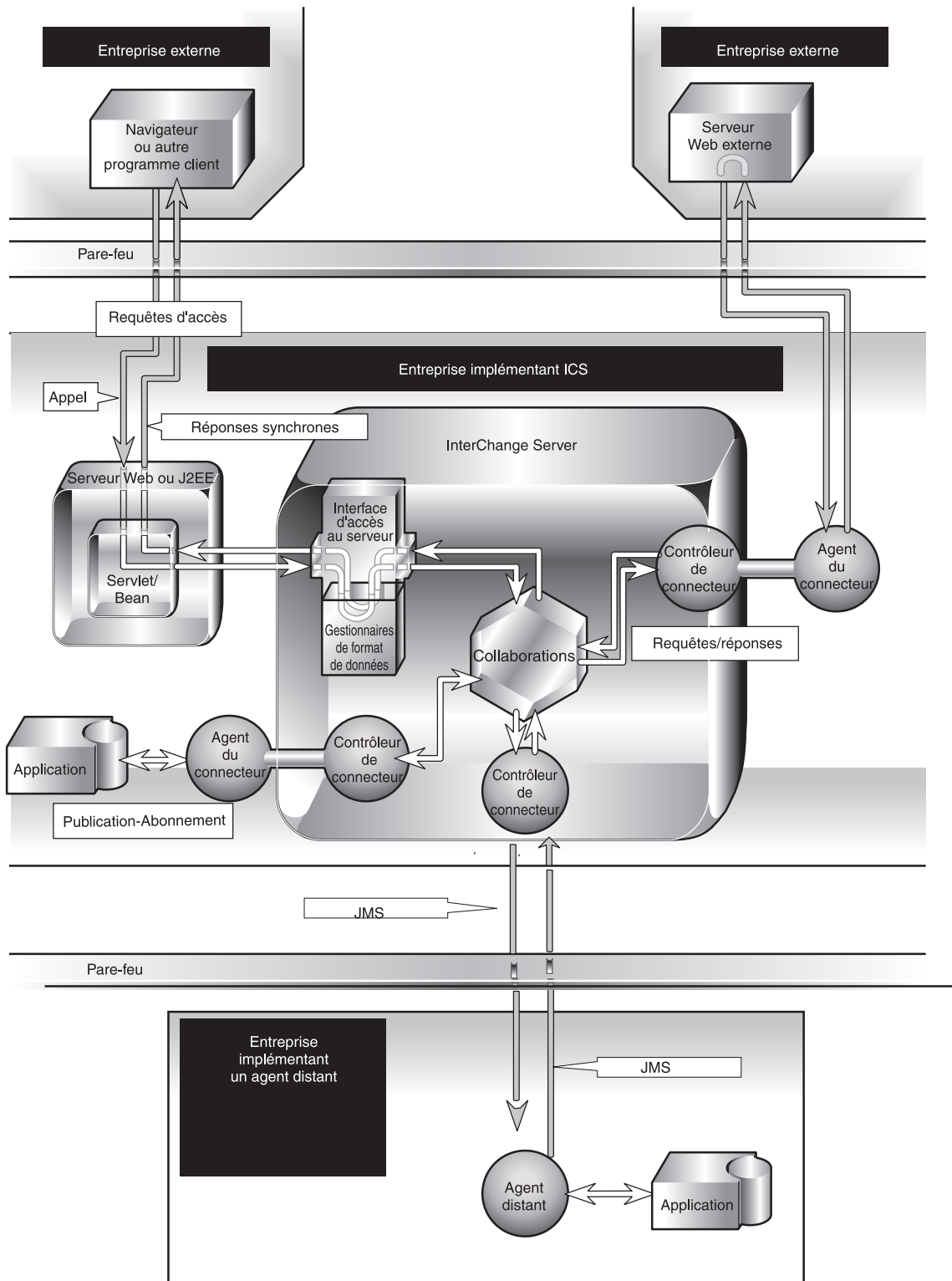


Figure 6. Flux des données de gestion

La figure 7 présente certains types de données qui sont échangés selon une partie des possibilités de chemins de données.

Les nombres indiquent la séquence spécifique d'une seule possibilité de chemin de données, (dans cet exemple, à partir d'un navigateur Web externe vers les

collaborations par l'intermédiaire de l'interface d'accès au serveur, puis, une fois le traitement métier réalisé (par la collaboration, vers un serveur Web externe).

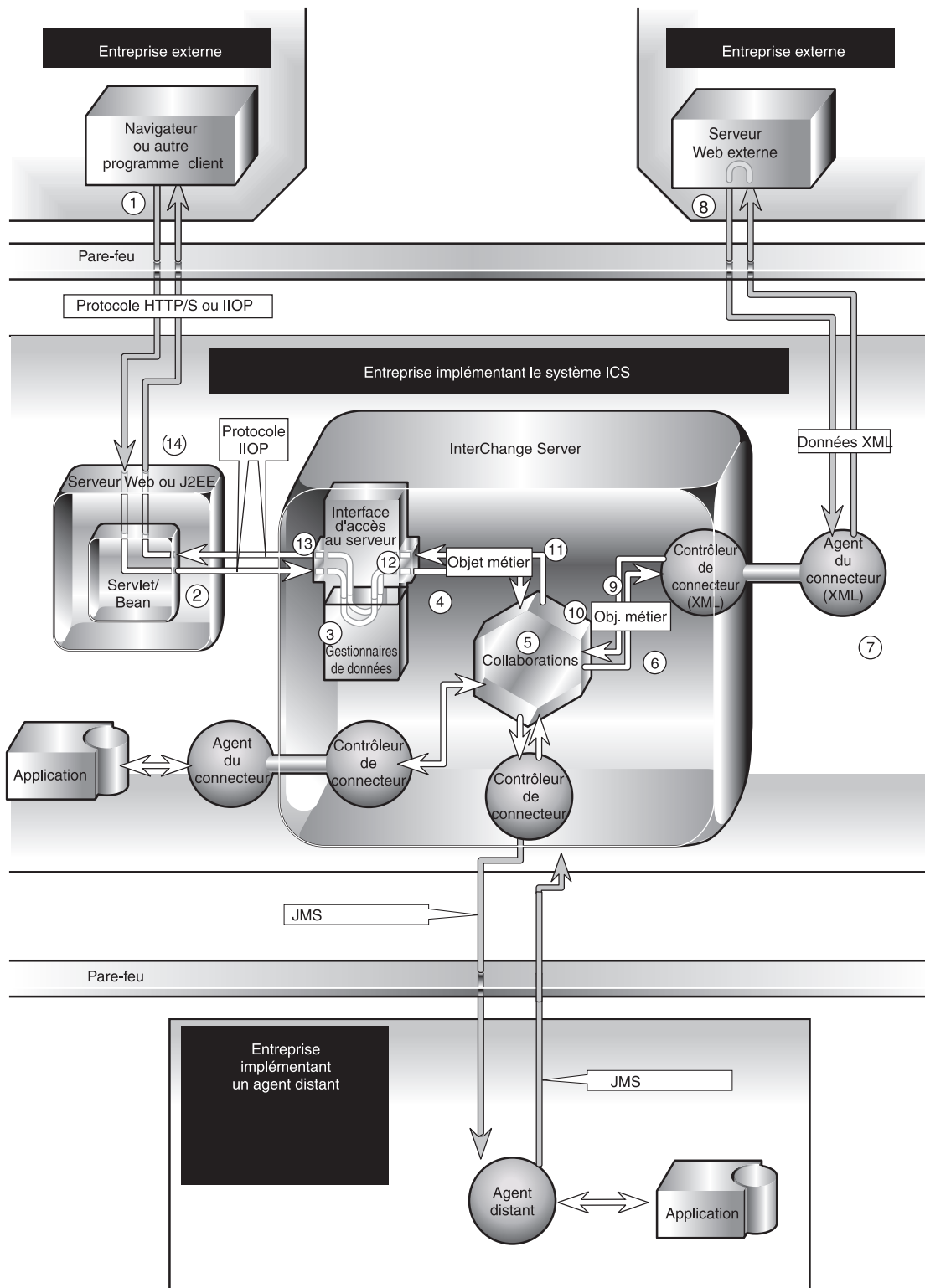


Figure 7. Exemple de flux de données de gestion

Dans l'exemple de la figure 7, le processus suivant est représenté :

1. Un programme client communique avec un serveur pour envoyer une requête d'accès à une collaboration. Le programme client pourrait être un navigateur Web qui communique avec un serveur Web selon le protocole HTTP ou HTTPS, ou un client J2EE effectuant un appel de méthode sur un serveur d'applications utilisant soit RMI soit IIOP.
2. L'appel est géré par le composant approprié côté serveur (un servlet dans le cas d'un serveur Web, et un EJB dans le cas d'un serveur d'applications J2EE) qui aura été configuré pour communiquer avec le système IBM WebSphere InterChange Server. Le composant spécifie l'objet à créer et la collaboration qui sera déclenchée par l'appel, puis il envoie l'appel à l'interface d'accès au serveur.
3. L'interface d'accès au serveur reçoit l'appel et le transfère au gestionnaire de données approprié (dans cet exemple, un gestionnaire de données XML). Le gestionnaire de données transforme les données en objet métier et les renvoie à l'interface d'accès au serveur.
4. L'interface d'accès au serveur transfère l'objet métier à la collaboration spécifiée.
5. La collaboration : (a) Exécute son processus métier sur l'objet métier, et (b) retourne de façon synchrone un objet métier à l'interface d'accès au serveur, qui sera remis au navigateur client ou à l'application cliente d'origine. L'objet métier peut contenir des données de gestion résultant du traitement, ou il peut s'agir d'un avis d'exception.
6. Dans cet exemple, la logique applicative de la collaboration lui indique qu'il faut envoyer un objet métier en tant que requête auprès d'un connecteur de technologies qui sera utilisé pour envoyer les données via Internet vers un serveur Web externe.
Encore dans un autre scénario, la collaboration pourrait au contraire envoyer un objet métier au connecteur d'une application au lieu de l'envoyer à un connecteur de technologies. L'application peut résider localement ou non, si la technologie d'agents distants est utilisée, l'application et son agent du connecteur pourraient résider à distance du concentrateur, via Internet.
7. Dans cet exemple, le connecteur XML transforme l'objet métier en document XML et l'envoie au serveur Web.
8. Le serveur Web envoie une réponse au connecteur XML.
9. Le connecteur XML transforme la réponse en objet métier et l'envoie à la collaboration.
10. La collaboration exécute des processus métier sur cet objet métier.
11. La collaboration envoie l'objet métier à l'interface d'accès au serveur.
12. L'interface d'accès au serveur envoie l'objet métier au gestionnaire de données approprié, et les données lui sont retournées dans ce format.
13. L'interface d'accès au serveur utilise le protocole IIOP pour envoyer les données au servlet.
14. Le servlet utilise le protocole HTTP ou HTTPS pour envoyer les données au navigateur Web ou toute autre entité d'origine.

Connecteurs

Les connecteurs font partie de la livraison avec le produit IBM WebSphere Business Integration Adapters. Un connecteur fournit des services de conversion distribués pour le système IBM WebSphere InterChange Server, transférant les données entre les collaborations et soit :

- Une application

- Une entité par programmation - un serveur Web distant, par exemple - qui comprend une norme de technologie, telle que le XML, qui est géré par un connecteur

Dans une implémentation InterChange Server, un connecteur a une structure répartie :

- Le **contrôleur de connecteur** interagit directement avec les collaborations et s'exécute comme un composant dans le processus InterChange Server.
- Une **structure à connecteur client** s'exécute de façon indépendante sur InterChange Server et, avec un composant propre à l'application, interagit directement avec une application ou tout autre entité par programmation. Dans ce guide, la structure à connecteur client et le composant propre à l'application sont assemblés dans une seule et même notion appelée **agent du connecteur**.

Les deux parties d'un connecteur peuvent fonctionner sur un même système ou sur deux systèmes différents. Le contrôleur de connecteur fonctionne en tant qu'élément d'InterChange Server et réside donc sur ce système. Cependant, l'agent du connecteur peut résider sur tout système à partir duquel il peut communiquer à la fois avec ses applications et le contrôleur de connecteur.

Communication du connecteur avec les applications

Il y a un connecteur pour chaque version d'une application. Chaque connecteur est unique, car il communique avec son application selon les interfaces de l'application. Lorsqu'il existe une interface de programme d'application (API), le connecteur peut l'utiliser. Cependant, le connecteur d'une application sans API peut utiliser quelque méthode que ce soit fournie par l'application, telle que les exits utilisateur ou les courriels.

Pour détecter les événements d'applications concernant les collaborations, le connecteur interroge l'application ou utilise le mécanisme d'avis de rappel d'événement de l'application, le cas échéant. Un connecteur peut également interagir avec une application sous les commandes d'une collaboration, ou pour vérifier le résultat de ses requêtes précédentes.

Liaisons entre les éléments

Pour exécuter un processus métier, la collaboration peut communiquer avec des connecteurs, avec d'autres collaborations et avec des processus externes desquels elle reçoit des requêtes d'accès via l'interface d'accès au serveur. Une **liaison** est utilisée pour établir des communications entre la collaboration et ces éléments une fois la collaboration configurée.

Liaison d'un déclencheur

Pour permettre le déclenchement des processus métier d'une collaboration, vous devez, lors de la configuration, lier la collaboration à un élément qui va générer l'événement déclencheur ou l'appel déclencheur.

Une liaison s'effectue entre la collaboration et tout élément qui va participer au processus métier de la collaboration, mais seul un élément peut être lié en tant que déclencheur.

Définition des liaisons pour les événements

Lorsque vous configurez une collaboration sur votre site pour qu'elle soit déclenchée par des événements, vous devez la lier à un connecteur qui soit en

mesure de publier l'événement déclencheur. Par exemple, vous pourriez spécifier que l'événement Employé.Supprimer provient du connecteur PeopleSoft.

Pour étudier de façon plus approfondie cette relation, souvenez-vous que le connecteur est constitué d'un contrôleur de connecteur (qui, comme la collaboration, comprend la structure à connecteur client ainsi qu'un composant propre à l'application, et est séparé d'InterChange Server). Le contrôleur de connecteur gère les informations relatives à la liaison et fournit à son agent du connecteur une liste d'événements auxquels les collaborations se sont abonnées.

Lorsqu'une opération pertinente de l'application se produit, l'agent du connecteur publie les événements sur cette liste à destination du contrôleur de connecteur. L'agent du connecteur envoie un événement au contrôleur de connecteur sans rien savoir sur la destination finale de cet événement dans la collaboration.

Le contrôleur de connecteur est par conséquent un intermédiaire entre l'agent du connecteur et la collaboration, comme cela est illustré dans la figure 8.

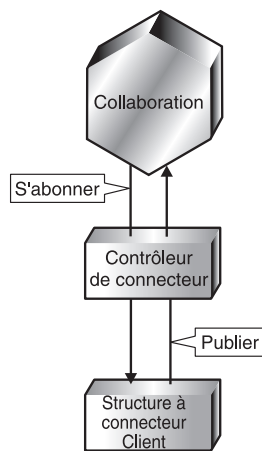


Figure 8. Un connecteur fournissant un événement déclencheur

De multiples collaborations peuvent s'abonner à un même événement. Lorsque le contrôleur de connecteur publie l'événement, il peut le faire simultanément à tous les abonnés.

Liaison pour recevoir des requêtes d'accès

Au lieu de relier la collaboration à un connecteur pour le déclenchement, vous pouvez spécifier que la collaboration recevra des requêtes d'accès à partir de processus externes, qui agiront comme des déclencheurs.

Liaison des destinations

De la même façon que vous liez les collaborations aux éléments de déclenchement, vous devez également lier les collaborations aux éléments de destination avec lesquels les collaborations vont engager des interactions requête/réponse. Les éléments de destination peuvent être soit des connecteurs soit d'autres collaborations. Une collaboration unique peut être liée à de multiples éléments de destination.

Objets métier

Les collaborations et les connecteurs interagissent en envoyant et en recevant des **objet métier** via InterChange Server.

Un objet métier reflète une entité de données : un ensemble de données qui peuvent être traitées en tant qu'unité opérationnelle. Par exemple, une entité de données peut être l'équivalent d'un formulaire, avec tous ses champs. Le formulaire pourrait par exemple être utilisé dans une application, ou sur le Web, pour contenir des informations métier à propos des clients, des employés ou des factures.

Les objets métier sont placés dans la mémoire cache pendant l'exécution de la collaboration pour un accès rapide, et enregistrés en mémoire à l'état de transaction persistante afin d'offrir un service fiable de reprise sur incident, de récupération amont et de nouvelle exécution des collaborations après un redémarrage du serveur en cas d'incident.

Le système IBM WebSphere InterChange Server crée des objets métier qui reflètent les informations contenues dans les entités. Dans ce manuel, il est souvent fait référence à une entité de données en fonction du type d'informations métier qu'elle contient, par exemple : une *entité employé* ou une *entité client*.

Cette section offre une première approche des objets métier. Plus loin dans ce guide, un chapitre fournit davantage d'informations sur les objets métier et leur contenu, et la façon dont le système IBM WebSphere InterChange Server gère et traite ces objets métier.

Rôles d'un objet métier

Un objet métier peut agir en tant qu'événement, requête ou réponse.

Événement

Un objet métier peut rapporter l'occurrence d'un **événement d'application**, c'est-à-dire une opération qui a modifié une entité de données dans une application. L'événement d'application pourrait être la création, la suppression ou le changement de la valeur de cette collecte de données. Lorsqu'un connecteur détecte un événement d'application et envoie un objet métier à la collaboration intéressée, le rôle de l'objet métier est de représenter l'événement, cet objet métier se fait donc appeler **événement** dans le système IBM WebSphere InterChange Server.

Par exemple, un connecteur pourrait interroger une application sur ses nouveaux employés de la part d'une collaboration. Si l'application crée une nouvelle entité employé, le connecteur envoie un objet métier événement à la collaboration.

Requête

Les requêtes sont en général créées de l'une de ces deux façons :

- Une collaboration peut envoyer un objet métier en tant que **requête** à un connecteur, demandant au connecteur d'insérer, de modifier, de supprimer ou d'extraire des données d'une application. Par exemple, dans la collaboration de service de facturation illustrée dans la figure 5, la collaboration envoie deux objets métier aux connecteurs, l'un pour extraire un contrat et l'autre pour créer une facture. Tous deux sont des requêtes.

- L'interface d'accès au serveur peut envoyer un objet métier en tant que requête auprès d'une collaboration, si cette collaboration a été conçue ou personnalisée pour accepter l'instruction Extraire (Retrieve) en tant que déclencheur.

Réponse

Lorsqu'un connecteur termine le traitement d'une requête, il retourne généralement une **réponse**. Par exemple, lorsqu'un connecteur reçoit une demande d'extraction des données relatives à un employé, à partir d'une application, il envoie un objet métier contenant les données de l'employé.

La structure d'un objet métier

Un objet métier est une unité auto-descriptive qui contient un type (son nom), des instructions de traitement (une instruction), et une donnée (valeurs d'attribut). La figure 9 représente un exemple d'objet métier simple, avec son type, son instruction et ses valeurs d'attribut.

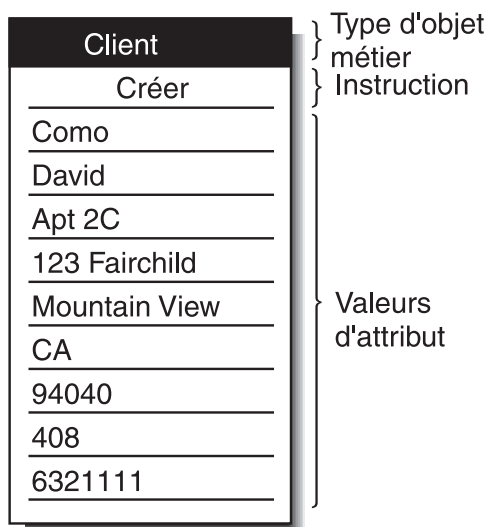


Figure 9. Composants d'un objet métier

La section suivante décrit ces composants.

Type d'objet métier

Chaque objet métier a un nom de type qui l'identifie dans le système IBM WebSphere InterChange Server. Ce type est défini par la définition de l'objet métier. Par exemple, le type pourrait être Client, Employé, Élément ou Contrat.

Instructions des objets métier

Un objet métier spécifie une action en relation avec les valeurs d'attribut. L'instruction peut indiquer différents types d'actions, en fonction du rôle de l'objet métier. Le tableau 1 répertorie les trois rôles de l'objet métier et décrit la signification de l'instruction pour chaque rôle de l'objet métier.

Tableau 1. Signification des instructions des objets métier

Rôle de l'objet métier	Signification de l'instruction
Événement	Décrit ce qui s'est passé dans une application. Par exemple, dans un événement, l'instruction Créer indique que l'application source a créé une nouvelle entité de données.

Tableau 1. Signification des instructions des objets métier (suite)

Rôle de l'objet métier	Signification de l'instruction
Requête	Indique au connecteur comment il doit interagir avec l'application afin de traiter l'objet métier. Par exemple, l'instruction Actualiser est une requête à l'attention du connecteur afin qu'il mette à jour l'entité données.
Réponse	Fournit les résultats d'une requête précédente. Par exemple, dans une réponse, le verbe Extraire indique que le connecteur a obtenu les valeurs d'attribut de la part de l'application.

Remarque : La convention d'attribution de nom veut que l'on utilise le format *type-d'objet-métier.instruction* pour indiquer un type d'objet métier particulier avec une instruction particulière. Par exemple, Client.Créer est un objet métier Client avec l'instruction Créer.

Valeurs d'attribut d'un objet métier

Un objet métier contient des **valeurs d'attribut** qui représentent des zones de données associées avec l'entité de données, telles que Nom, Prénom, ID Employé, ou Statut de facturation.

Certains attributs, au lieu de contenir des données, contiennent des **objets métier enfants** ou **modules d'objets métier enfants**. La figure 10 illustre la structure d'un objet métier Contrat. Les informations relatives à la ligne article dans le contrat sont un module d'objets métier enfants.

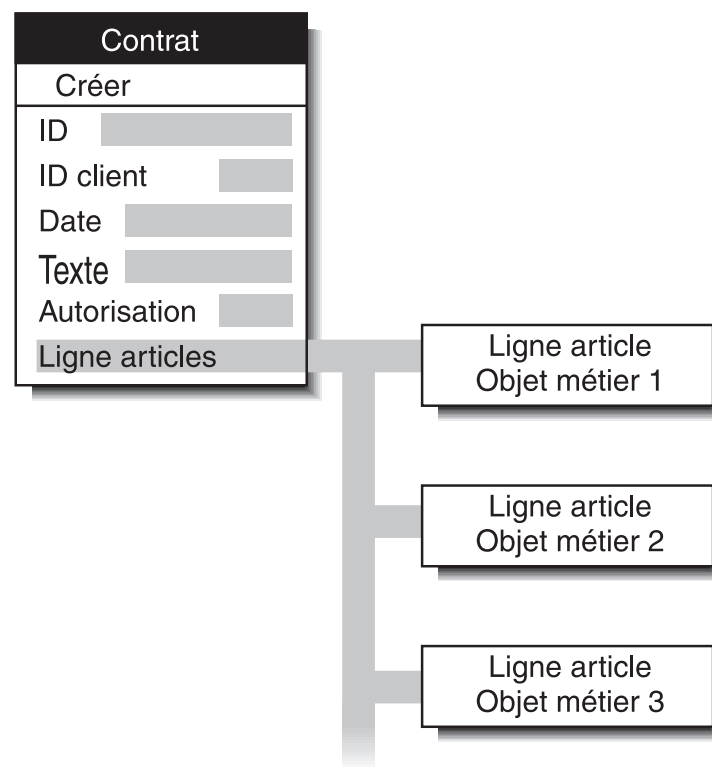


Figure 10. Objets métier avec objets métier enfants

Un objet métier qui contient des objets métier enfants ou des modules d'objets métier enfants est un **objet métier hiérarchique**. Ceux dont les attributs ne contiennent que des données sont des **objets métier à plat**.

Objets métier propres à une application et génériques

Le système IBM WebSphere InterChange Server comprend deux sortes d'objets métier : propres à l'application et génériques.

- Un **objet métier propre à l'application** reflète des attributs d'entités de données ainsi qu'un modèle de données d'une application spécifique ou d'une entité par programmation.
- Un **objet métier générique** contient un ensemble d'attributs liés à l'activité qui sont communs à un large ensemble d'applications, sans les limites imposées par le modèle de données d'une application spécifique.

Lorsque l'agent du connecteur (par l'intermédiaire de son composant propre à l'application) détecte un événement d'application tel qu'une mise à jour, il extrait l'entité de données correspondantes de l'application et la transforme en **objet métier propre à l'application**.

Remarque : Lorsque ce document fait référence à un objet métier dont le nom inclut un nom d'application, tel que Clarify_Contact ou Oracle_Client, il s'agit d'un objet métier propre à une application. Un objet métier Clarify_Contact, par exemple, contient l'ensemble des informations que l'application Clarify stocke sur ce contact. Dans une autre application, une entité contact peut stocker un ensemble d'informations quelque peu différent, stocker les informations dans un ordre ou un format différents, ou porter un nom différent.

Une fois que l'agent du connecteur a construit un objet métier propre à l'application, il envoie l'objet métier au contrôleur de connecteur dans InterChange Server.

Le contrôleur de connecteur échange des objets métier entre des collaborations et l'agent du connecteur. Les collaborations sont en général neutres vis-à-vis des applications, l'objet métier que le contrôleur de connecteur échange avec une collaboration doit donc être un **objet métier générique**. L'utilisation des objets métier génériques favorise la réutilisation de la collaboration, car sa logique métier n'est pas limitée à des versions spécifiques d'applications spécifiques.

Remarque : Les noms des objets métier génériques n'incluent ni un nom de société, ni un nom de produit. Contact, Employé, et Client sont des exemples de noms génériques.

La figure 11 montre l'endroit où les deux catégories d'objets métier s'insèrent dans le système IBM WebSphere InterChange Server : la collaboration interagit à l'aide d'objets métier génériques, et l'agent du connecteur prend en charge des objets métier conçus pour des applications spécifiques.

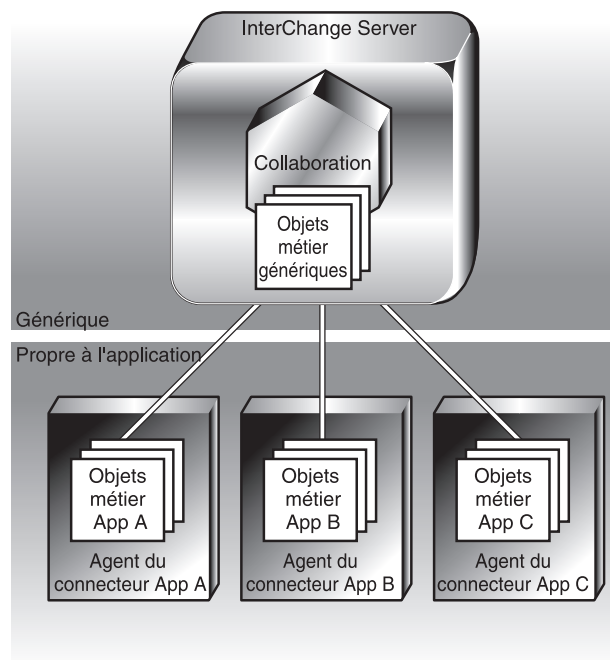


Figure 11. Objets métier génériques et propre à des applications

Vous pouvez utiliser le même connecteur pour exécuter plusieurs types de collaborations, si le connecteur prend en charge les objets métier utilisés par ces collaborations.

Le mappage des données

La figure 11 montre différents types d'objets métier au niveau de chaque application et des collaborations. Le système IBM WebSphere InterChange Server doit par conséquent convertir les objets métier en sortes de formulaires afin de pouvoir envoyer des événements et des données à travers les applications et les collaborations. Le **mappage de données** est le processus de conversion des objets métier d'un type vers un autre type. Le mappage de données est requis à chaque fois que le système IBM WebSphere InterChange Server envoie des données entre une source et une cible qui ne partagent pas exactement le même modèle de données.

Contrairement aux solutions d'intégration d'applications personnalisées qui mappent les données directement d'une application à l'autre, les collaborations InterChange Server utilisent généralement l'objet métier générique entre les modèles de données propres aux applications. L'objet métier générique sert d'ensemble de données inter-application commun. Si vous changez d'application à l'avenir, vous n'aurez qu'à utiliser un nouveau connecteur et à mapper le nouvel objet métier propre à l'application vers l'objet métier générique. Les collaborations continuent alors à fonctionner comme auparavant.

A chaque fois qu'une collaboration transfère des objets métier à travers des applications dissemblables, le mappage transforme l'objet métier en direction et en provenance de l'ensemble de données communes. La transformation des objets métier s'effectue :

- À partir des objets métier propres à une application vers des objets métier génériques lorsqu'ils passent des connecteurs vers des collaborations
- À partir des objets métier génériques vers des objets métier propres à une application lorsqu'ils passent des collaborations vers des connecteurs

Par exemple, dans une collaboration qui synchronise des données Clarify_BusOrg avec des données SAP_CustomerMaster, le mappage s'effectue deux fois :

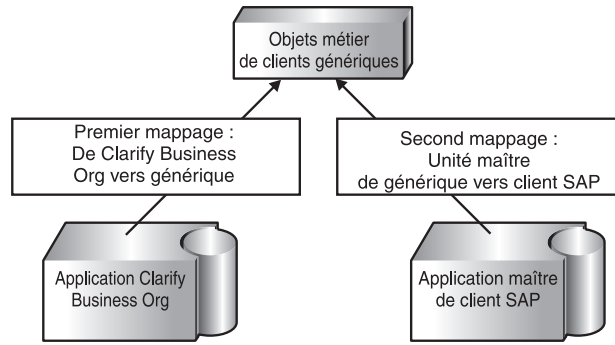


Figure 12. Transformations du mappage

Chaque contrôleur de connecteur gère le mappage des objets métier qui passent entre ses agents du connecteur et InterChange Server. Pour réellement effectuer le mappage de données, cependant, le système appelle les outils de mappage—Map Designer et Relationship Designer. Ces outils vous permettent de créer et de modifier des spécifications de mappage détaillées et d'exécuter le mappage lors de la phase d'exécution.

Un contrôleur de connecteur appelle la fonction de mappage lorsqu'il reçoit des objets métier qui requièrent un mappage. La figure 13 illustre la commande d'appel de mappage en provenance du contrôleur de connecteur.

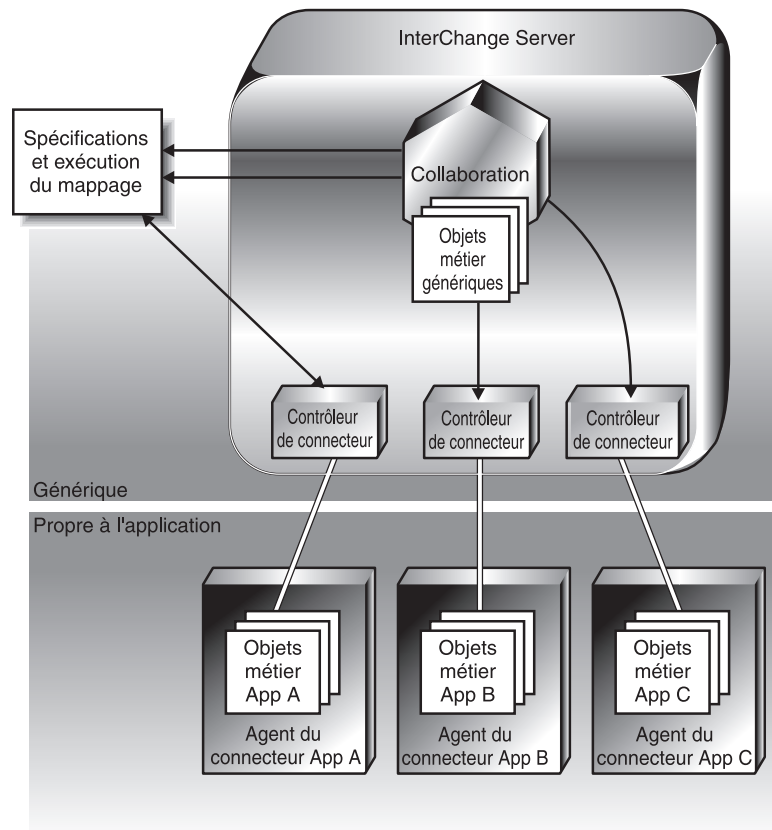


Figure 13. Contrôle et exécution du mappage

InterChange Server

InterChange Server est une structure d'exécution multiprocessus en Java pour les collaborations. InterChange Server s'exécute dans sa propre machine virtuelle Java (JVM) InterChange Server s'exécute sur des systèmes Windows et UNIX. Cette section décrit les services et fonctionnalités suivants d'InterChange Server :

- «Service de gestion des événements»
- «Contrôleur de connecteur» à la page 26
- «Référentiel» à la page 26
- «Service de connectivité de la base de données» à la page 26
- «Pools de connexion à des bases de données» à la page 27
- «Haute disponibilité» à la page 27
- «Service de transaction» à la page 27
- «Fonction de reprise sur incident» à la page 29

Service de gestion des événements

InterChange Server stocke de façon continue chaque objet métier qu'il reçoit pendant l'exécution d'une collaboration. Ceci permet à InterChange Server de récupérer après un arrêt imprévu ou l'échec d'une collaboration sans perdre de notification d'événement ni d'appel.

Contrôleur de connecteur

Un contrôleur de connecteur est une interface entre le côté client d'un connecteur et InterChange Server. Un contrôleur de connecteur achemine les objets métier lors de leur traversée du système IBM WebSphere InterChange Server en reliant le côté client des connecteurs aux collaborations et en gérant le processus de mappage.

Par l'intermédiaire d'un contrôleur de connecteur, un administrateur peut :

- Suivre les interactions entre InterChange Server et l'agent du connecteur.
- Activer et désactiver les interactions entre InterChange Server et l'agent du connecteur.
- Spécifier le type de mappage à effectuer pour chaque objet métier qui arrive et qui part de l'agent du connecteur.

Référentiel

InterChange Server gère les informations sur la configuration et la définition de chaque objet dans un stock de données persistant appelé le **référentiel InterChange Server**, qui est constitué d'un ensemble de tables dans une base de données relationnelle. Les tables stockent les informations sur la définition et la configuration des objets sous la forme de documents XML.

Service de connectivité de la base de données

Le **service de connectivité de la base de données** gère les interactions entre InterChange Server et le référentiel. Le service de connectivité de la base de données interagit avec le référentiel par l'intermédiaire d'une API (interface de programme d'application) JDBC (Java Database Connectivity), comme l'illustre la figure 14.

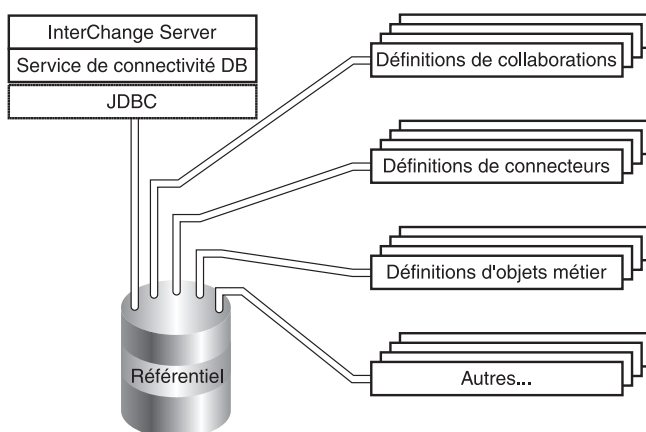


Figure 14. Le service de connectivité de la base de données et le référentiel

Remarque : Certaines bases de données disposent des pilotes de l'interface universelle de connexion aux bases de données (ODBC), mais n'ont pas de pilotes JDBC natifs. Pour ces bases de données, vous pouvez utiliser un pont JDBC-ODBC.

Une implémentation InterChange Server prend en charge de nombreuses bases de données d'autres fournisseurs et de nouvelles bases de données sont constamment en cours de certification. Pour la liste actuelle des bases de données prises en charge, reportez-vous au manuel *System Installation Guide (Windows ou UNIX)*.

Pools de connexion à des bases de données

Vous pouvez utiliser l'outil System Manager du système IBM WebSphere InterChange Server pour définir les pools de connexion à une base de données dans InterChange Server. Les pools de connexions à une base de données définis par l'utilisateur permettent aux développeurs d'accéder directement aux bases de données relationnelles à partir d'une collaboration ou d'une mappe. Cette fonctionnalité permet la prise en charge des éléments suivants :

- Gestion automatisée du cycle de vie des connexions à une base de données
- API simplifiées pour les instructions SQL et l'exécution des procédures stockées
- Transaction concaténation-séparation gérée par conteneur

Haute disponibilité

Le système IBM WebSphere InterChange Server peut être configuré pour fournir une à haute disponibilité à InterChange Server (ICS).

Sur les systèmes Windows, la configuration à haute disponibilité est obtenue avec le logiciel Cluster Server de Microsoft(MSCS). La configuration à haute disponibilité nécessite deux serveurs en grappe certifiés Microsoft. Les deux serveurs sont installés avec la même configuration système InterChange Server et sont désignés en tant que système en grappe dans le logiciel MSCS. L'une des machines est configurée dans MSCS en tant que serveur primaire (le serveur actif jusqu'à ce qu'une panne survienne), et l'autre en tant que serveur de secours. Les deux machines partagent un nom de grappe et une adresse IP de grappe que les processus externes utilisent pour accéder au serveur actif. Le serveur primaire ainsi que le serveur de secours ont accès à un système de stockage RAID partagé, qui est contrôlé par le serveur actif.

Lorsqu'une panne du matériel est détectée, la configuration à haute disponibilité ferme le système InterChange Server (ICS), effectue une migration automatique, et redémarre ICS (ainsi que le logiciel tiers et les connecteurs à haute disponibilité) sur le système de secours en grappe. Le serveur de secours en grappe devient le serveur actif, s'appropriant le nom de grappe et l'adresse IP du serveur primaire, et prenant automatiquement en charge les traitements système jusqu'à ce que vous réparez le serveur primaire et que vous réinitialisiez la reprise (c'est-à-dire que vous rebasculiez manuellement les traitements vers le serveur primaire).

Pour une présentation générale de l'environnement à haute disponibilité, consultez le manuel *System Administration Guide*. Pour des informations complémentaires sur la façon de configurer ICS pour une utilisation dans un environnement à haute disponibilité, consultez le manuel *System Installation Guide*.

Service de transaction

Chaque application de solution d'intégration rencontre des risques lors du transfert de données d'une application vers une autre. Lorsque les données quittent l'environnement protégé d'une application et de sa base de données pour traverser un réseau vers une autre application, un nombre incalculable de problèmes peuvent survenir. Par exemple :

- Le réseau pourrait tomber en panne.
- L'application cible pourrait se planter avant d'avoir reçu les données.
- Une erreur pourrait survenir dans l'application cible avant qu'elle ait pu traiter les nouvelles données.

Prenons par exemple une collaboration qui implique trois applications : ressources humaines, paie, et comptabilité des coûts de production. Chaque application stocke le salaire des employés. La figure 15 est une vue fonctionnelle de la logique applicative inter-application de la gestion de l'augmentation du salaire d'un employé.

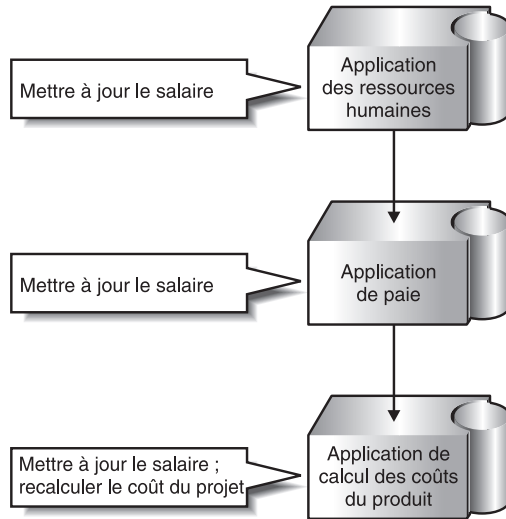


Figure 15. Trois applications en train de traiter le salaire des employés

Lorsqu'un employé obtient une augmentation, une personne saisit un nouveau salaire dans l'application des ressources humaines. La collaboration met automatiquement à jour le salaire de l'employé dans l'application de paie, puis ajuste le coût du projet dans l'application de comptabilité des coûts de production.

Si une erreur système empêche la mise à jour du salaire d'atteindre l'application de comptabilité des coûts de production, il peut en résulter une incohérence. La valeur du salaire de l'employé est désormais différente dans les applications, et le calcul du coût de production est par conséquent incorrect.

Traditionnellement, il fallait vérifier les données de façon méticuleuse par recoupement ou bien les données restaient erronées jusqu'à ce que quelqu'un découvre le problème. Cependant, le système IBM WebSphere InterChange Server offre un meilleur niveau de service avec des contrôles stricts vous permettant d'avoir toute confiance dans les données de l'entreprise, grâce à des services qui peuvent exécuter une collaboration comme si c'était une sorte de **transaction**.

Collaborations transactionnelles

Des qualités transactionnelles sont souhaitables et possibles pour des collaborations dans lesquelles la cohérence des données entre les applications est importante. Comme pour les autres transactions, une collaboration transactionnelle implique un certain nombre d'étapes. Lorsqu'une erreur survient, InterChange Server peut annuler chaque étape terminée, en effectuant une récupération sous forme de transaction.

Cependant, les collaborations diffèrent des transactions traditionnelles de plusieurs manières :

- Les actions des collaborations sont réparties, et il n'existe aucun contrôle centralisé sur les bases de données participantes.

- Les collaborations qui répondent à des événements (comme dans le modèle publication-abonnement) sont des collaborations longue durée : elles s'exécutent de façon asynchrone, car le fait d'isoler les données des applications aurait un impact défavorable sur les utilisateurs des applications.
- Les applications sauvegardent les modifications de données effectuées par les collaborations, procurant ainsi une forme de durabilité inter-application répartie. Cependant, si une collaboration doit effectuer une récupération amont, il sera peut-être nécessaire d'annuler des opérations sauvegardées auparavant.

Par conséquent, les techniques utilisées par InterChange Server pour prendre en charge les collaborations transactionnelles sont différentes de celles qui prennent en charge les transactions traditionnelles. Les niveaux de transaction associés aux collaborations définissent la rigueur avec laquelle InterChange Server applique la sémantique transactionnelle.

Fonction de reprise sur incident

Une implémentation InterChange Server dispose de fonctionnalités permettant de réduire la durée nécessaire au réamorçage d'ICS après une panne, afin qu'ICS soit disponible pour d'autres tâches avant que la récupération de tous les flux soit terminée, et pour contrôler la nouvelle soumission des événements qui ont échoué :

- Reprise sur incident asynchrone
InterChange Server n'attend pas la récupération des collaborations et des connecteurs pour achever son réamorçage : les collaborations et les connecteurs sont autorisés à récupérer de façon asynchrone après le réamorçage d'InterChange Server. Ceci permet d'utiliser les outils de dépannage de System Manager, tels que System Monitor et la boîte de dialogue de gestion des flux non résolus, tandis que les connecteurs et les collaborations sont en train de se reconstituer.
- Reprise sur incident différée
L'utilisation de cette fonction est facultative et sa configuration s'effectue via les propriétés de l'objet de collaboration. Si vous activez cette fonction pour une collaboration, en cas de panne de l'ICS, les flux de la collaboration en cours de traitement sont différés après le réamorçage du serveur, libérant ainsi l'utilisation de la mémoire associée à ces flux. Après le réamorçage du serveur, vous pouvez utiliser la boîte de dialogue de gestion des flux non résolus pour transmettre de nouveau les événements.
- Maintenir des interventions sur appel à l'état 'en transit'
Il est possible que vous souhaitiez empêcher une reprise sur incident de soumettre de nouveau toutes les interventions sur appel qui étaient en transit au moment de la panne, afin d'éviter l'éventuel envoi par une collaboration non transactionnelle d'événements en double à une application cible. Pour ce faire, il faut configurer la collaboration (avant la panne du serveur) afin qu'elle maintienne tout événement d'intervention sur appel dans l'état 'en transit' lorsque survient une reprise sur incident consécutive à une panne. Pendant la récupération d'InterChange Server, les flux qui étaient en train de traiter les interventions sur appel restent à l'état 'en transit', et vous pouvez utiliser la boîte de dialogue de gestion des flux non résolus pour examiner les flux non résolus et contrôler s'il faut les soumettre de nouveau et quand, le cas échéant.
- Fonctions de distribution des événements garantis
Pour les connecteurs compatibles JMS (les connecteurs qui utilisent JMS comme mécanisme de transport), les fonctionnalités suivantes peuvent être utiles pour la distribution des événements garantis en cas de reprise sur incident :

- Événements gérés par conteneur
La fonction d'événement géré par conteneur est valide pour les connecteurs compatibles JMS qui utilisent un stockage d'événements JMS. Cela permet, en cas de panne et de reprise sur incident d'un système, à un événement (qui était en cours de traitement entre le stockage d'événement et la structure de connecteur) de n'être reçu qu'une fois par la structure de connecteur. Cette fonction est facultative et configurée dans les propriétés du connecteur ; elle n'est utilisée qu'avec les connecteurs qui utilisent JMS comme mécanisme de transport.
- Élimination des événements en double
La fonction d'élimination des événements en double, valable pour les connecteurs compatibles JMS, utilise des identificateurs d'événement uniques dans le code propre à l'application du connecteur, afin de garantir que les événements ne vont pas être distribués deux fois à la file d'attente de distribution. Cette fonction est facultative et configurée dans les propriétés du connecteur ; elle n'est utilisée qu'avec les connecteurs qui utilisent JMS comme mécanisme de transport.

Infrastructure de transport des communications

L'architecture modulaire d'une implémentation InterChange Server forme un système réparti qui s'adapte à de nombreux types de configurations de sites. L'architecture de l'implémentation InterChange Server permet des interactions réparties entre de multiples machines sur un réseau, et une interaction répartie à travers les pare-feu Internet.

Distribution sur un réseau

Les interactions entre des composants InterChange Server répartis sur un réseau sont activées par le standard Corba et par des technologies de messagerie, y compris les systèmes natifs suivants : messagerie WebSphere MQ et Java Messaging Service (JMS).

Plusieurs configurations sont possibles. Le standard CORBA peut être utilisé pour des interactions requête/réponse et des communications administratives entre l'agent du connecteur et l'ICS, avec WebSphere MQ en natif qui réalise la distribution des événements dans des opérations publication-abonnement. Alternativement, Java Messaging Service peut être utilisé pour toutes les communications et les interactions entre un agent du connecteur et ICS.

Standard CORBA

L'architecture pour un intermédiaire commun dans les requêtes d'objet (**CORBA**) définit un ensemble de normes et d'interfaces pour la distribution d'objets sur un réseau. Le répartiteur de requêtes d'objets, ou Object Request Broker (**ORB**) est un ensemble de bibliothèques et de composants que les applications client et les serveurs d'objets utilisent pour communiquer. InterChange Server utilise le logiciel IBM Java ORB.

Le répartiteur de requêtes d'objets permet à InterChange Server d'être accessible à ses clients, à l'agent du connecteur et à l'administrateur du système. InterChange Server s'enregistre auprès du service des noms du répartiteur de requêtes d'objets, à partir duquel un client obtient les informations dont il a besoin pour trouver le serveur et commencer à interagir avec lui. Le client et le serveur effectuent des interactions objet-objet par l'intermédiaire du langage de définition de l'interface du répartiteur de requêtes d'objets (IDL). Au niveau du transport, le client et le

serveur communiquent via le protocole IIOP (Internet Inter-ORB protocol). Les communications à base de répartiteurs de requêtes d'objets sont généralement utilisées pour les objectifs suivants :

- L'interface d'accès au serveur utilise les communications à base de répartiteurs de requêtes d'objets pour gérer les appels.
- Dans les interactions requête/réponse, les collaborations et les connecteurs utilisent les communications à base de répartiteurs de requêtes d'objets pour échanger leurs objets métier.
- Un agent du connecteur utilise les communications à base de répartiteurs de requêtes d'objets :
 - Lors du démarrage, lorsqu'il interagit avec InterChange Server pour obtenir sa configuration initiale.
 - En cours de fonctionnement, lorsqu'il reçoit des instructions du contrôleur de connecteur : indiquer le statut, faire une pause, s'arrêter ou reprendre.
- Les communications à base de répartiteurs de requêtes d'objets peuvent éventuellement être utilisées pour la distribution d'événements dans une interaction publication-abonnement.

Dans le protocole requête/réponse IIOP, soit la communication fonctionne, de façon immédiate, soit elle échoue. Les deux programmes doivent donc être en cours de fonctionnement pour communiquer. Cependant, pour les interactions requête/réponse dans une implémentation InterChange Server, vous pouvez utiliser la propriété d'un connecteur pour sélectionner le mode store-and-forward (stocker-faire suivre), et paramétrer la façon dont le contrôleur de connecteur doit répondre à une requête de collaboration dans une situation où l'agent du connecteur n'est pas disponible :

- Si vous configurez la propriété sur True (vrai), le contrôleur de connecteur ne manquera aucune requête de collaboration, même lorsque l'agent du connecteur sera indisponible. Une requête est bloquée jusqu'à ce que l'agent du connecteur redevienne opérationnel. La collaboration est par conséquent obligée d'attendre que l'agent du connecteur soit opérationnel pour pouvoir achever le flux du traitement de la requête.
- Si la propriété est configurée sur False (faux), le contrôleur de connecteur manquera toutes les requêtes de collaboration effectuées lorsque l'agent du connecteur est indisponible. Dans ce cas, la collaboration termine le traitement de la requête selon sa logique métier pour le traitement des requêtes qui ont échoué.

La figure 16 montre les composants du répartiteur de requêtes d'objets IBM Java ORB à utiliser avec InterChange Server. Il existe plusieurs composants pour les différents langages utilisés par les composants InterChange Server pour adresser leurs pilotes ORB, mais les composants C++ et Java communiquent les uns avec les autres comme un seul système. Le serveur de noms transitoire d'IBM fournit le service d'affectation de nom.

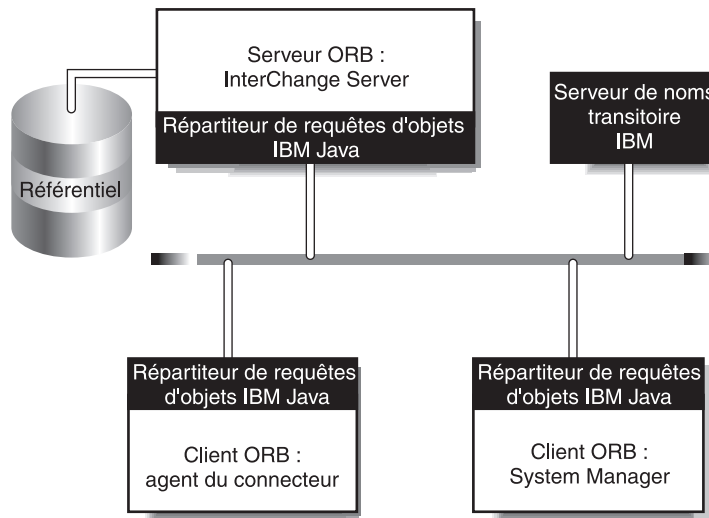


Figure 16. Les composants ORB dans le système WebSphere InterChange Server

Ces composants ORB ne doivent pas être séparés. Sur certains sites, tous ces composants doivent résider sur la même machine.

Les technologies de messagerie

La messagerie est une catégorie de communication dans lequel des programmes échangent des unités de données (messages) de façon discontinue et asynchrone. Les programmes qui utilisent la messagerie n'ont pas besoin d'établir de connexion ou d'attendre les messages : chaque programme envoie et reçoit de façon asynchrone des messages en interagissant avec le service de messagerie. Le service de messagerie fournit une distribution garantie, en stockant le message lorsque le programme cible n'est pas disponible et en réessayant jusqu'à ce que le programme soit disponible.

Les systèmes de messagerie pris en charge avec un environnement InterChange Server incluent à la fois les logiciels IBM WebSphere MQ messaging et Java Messaging Service (JMS) en natif. En général, une implémentation type utilise un système de messagerie pour la distribution des événements pour des interactions publication-abonnement, et utilise l'ORB pour des interactions requête/réponse. Cependant, il existe des circonstances dans lesquelles JMS est utilisé pour les deux types d'interactions.

Lorsque JMS est le mécanisme de transport de la distribution, la persistance des données peut être obtenue grâce à la fonction longue durée du processus métier. Lorsque cette fonction est utilisée, un processus initié par une requête sur une collaboration peut être mis en état d'attente avec une valeur de délai d'attente, afin de permettre au processus de reprendre lorsqu'il reçoit une réponse spécifique. L'utilisation de cette fonction exige que la fonction soit activée lors de la création du modèle de collaboration.

La distribution via Internet

Le système InterChange Server prend en charge les communications avec la couche de connexion sécurisée (SSL) pour distribuer les interactions contrôleur de connecteur/agent du connecteur à travers les pare-feu Internet.

L'implémentation est une relation à concentrateur en étoile, dans laquelle le concentrateur est un site sur lequel est installé un système IBM WebSphere

InterChange Server complet, et les postes distants sont des sites qui échangent des données avec le concentrateur à travers des pare-feu. Un site distant a besoin d'un agent du connecteur éloigné, mais n'a pas besoin d'un système IBM WebSphere InterChange Server complet.

Deux configurations sont possibles :

- WebSphere MQ avec SSL
- WebSphere MQ passe-système Internet utilisant HTTP/HTTPS

La sécurisation d'InterChange Server

Le système WebSphere Business Integration est un système réparti dont les données de gestion sensibles doivent être sécurisées. Les parties non autorisées ne doivent pas pouvoir consulter ni modifier les données pendant leur transfert ou leur stockage sur un disque. La sécurisation des données de tout système implique trois fonctions de sécurité qui sont l'authentification, l'intégrité des messages et la confidentialité. L'authentification implique la vérification de l'identité des utilisateurs. L'authentification régit le contrôle des accès basé sur les rôles, qui limite l'accès à des parties du système en fonction de l'identité de l'utilisateur et des autorisations qui lui ont été accordées. L'intégrité des messages garantit que les données n'ont pas été modifiées. La confidentialité permet d'assurer que seuls les utilisateurs autorisés peuvent voir les données. L'intégrité, la confidentialité et l'authentification peuvent être garanties par le chiffrement, tandis que le contrôle d'accès basé sur les rôles (RBAC) peut être obtenu grâce à la création d'un identificateur de l'utilisateur et d'un mot de passe.

Le chiffrement

Outre les trois fonctions de sécurité déjà citées, la sécurisation d'InterChange Server implique trois autres mécanismes de sécurité. Deux d'entre eux portent sur le codage des données via le chiffrement. Les deux approches de l'encodage des données sont le **chiffrement symétrique** et le **chiffrement asymétrique**. Le chiffrement symétrique implique l'encodage des données à l'aide d'un algorithme (ou clé) de chiffrement. Le même algorithme, ou clé, est utilisé pour chiffrer les données et les déchiffrer. Dans un système réparti, dans lequel un processus doit chiffrer les données et un autre processus doit les déchiffrer, la clé doit être partagée ou échangée de façon sûre, afin d'empêcher tout utilisateur non autorisé d'utiliser cette clé pour accéder aux données. Le partage d'une clé peut être une tâche difficile dans le contexte d'un chiffrement symétrique.

Le chiffrement asymétrique est communément utilisé pour partager ou échanger une clé. Cela implique de chiffrer et de déchiffrer les données avec une **paire de clés**. L'une des clés de la paire de clés est une **clé privée** qui est gardée secrète, et l'autre clé de la paire de clés est une **clé publique** qui est partagée et distribuée à d'autres. Les données chiffrées avec une clé publique ne peuvent être déchiffrées que par la clé privée correspondante de la paire de clés. L'expéditeur chiffre les données en utilisant la clé publique du réceptionnaire, afin que seul le réceptionnaire approprié puisse déchiffrer les données à l'aide de sa propre clé privée. Le chiffrement asymétrique est lent, il n'est donc pas utilisé pour chiffrer les données. Il est utilisé pour chiffrer et déchiffrer de façon asymétrique la clé secrète, qui est à son tour utilisée pour chiffrer symétriquement les données. Ce processus offre le meilleur compromis possible entre sécurité et vitesse car la confidentialité des données du système repose d'abord sur la solidité de la clé.

La clé privée peut également être utilisée pour créer une **signature numérique** pour les données, qui peut alors servir à vérifier l'identité de l'expéditeur et

l'intégrité des données envoyées. Pour créer une signature numérique, les données sont réduites à quelques lignes par un processus appelé **hachage** qui crée une **synthèse de message**. La synthèse de message est chiffrée avec la clé privée de l'émetteur, ce qui crée la signature numérique. La signature numérique est annexée aux données, et le tout est envoyé au réceptionnaire. Le réceptionnaire déchiffre la signature numérique dans la synthèse de message en utilisant la clé publique, qui vérifie la signature, car la clé publique ne peut déchiffrer que des données qui ont été chiffrées avec la clé privée, dont seul l'expéditeur dispose. Le réceptionnaire hache alors les données pour créer une synthèse de message, et la compare avec la synthèse de message qui a été envoyée. Si les deux synthèses de messages sont les mêmes, le réceptionnaire peut vérifier que les données n'ont pas été modifiées depuis qu'elles ont été signées par l'expéditeur.

InterChange Server offre les dispositifs de sécurité suivants pour protéger les données de gestion sensibles :

- Confidentialité complète (également appelée "sécurité complète"), les données sont sécurisées lorsqu'elles circulent à partir du processus d'adaptateur source, via InterChange Server, vers le processus d'adaptateur cible. La confidentialité complète utilise le chiffrement asymétrique et la sécurité dynamique pour l'authentification, l'intégrité et la confidentialité.
- Contrôle des accès basé sur les rôles, qui limite l'accès à des parties du système en fonction de l'identité de l'utilisateur et des autorisations qui lui ont été accordées. Le contrôle des accès basé sur les rôles couvre le contrôle d'accès et l'authentification.

Confidentialité complète

Les processus d'adaptateur et InterChange Server communiquent en s'envoyant des messages. Les messages sont bidirectionnels, InterChange Server et les processus d'adaptateur peuvent ainsi envoyer et recevoir des messages. Le protocole SSL offre une sécurité noeud à noeud ou au niveau de la liaison. Les messages peuvent être sécurisés lorsqu'ils voyagent entre les noeuds, mais pas lorsqu'ils sont stockés sur un disque dans un noeud. Les messages et les données sont alors en clair, visibles et modifiables par des utilisateurs non autorisés. La confidentialité complète peut sécuriser les messages de bout en bout, depuis leur noeud d'origine, à travers tous les éventuels noeuds intermédiaires, jusqu'au noeud de destination. Grâce au chiffrement asymétrique et à la sécurité dynamique, les messages en cours de transmission sont sécurisés, et ils sont également sécurisés lorsqu'ils sont dans les gestionnaires de files d'attente sur les disques, en attendant d'être traités. Les messages sont sécurisés selon des niveaux de sécurité variables, qui peuvent inclure le chiffrement et l'**authentification aux points d'extrémité**. L'authentification aux points d'extrémité survient lorsque les points d'extrémité de la communication, tels InterChange Server ou un adaptateur, prouvent qu'ils sont bien qui ils prétendent être grâce à leur signature numérique.

La sécurité asymétrique et dynamique

Dans un système réparti, les parties communiquant entre elles peuvent être à la fois des expéditeurs et des réceptionnaires. Lorsque la communication doit être sécurisée, et que les niveaux de sécurité sont les mêmes pour les messages entrants et sortants, il s'agit de sécurité symétrique. Ce type de sécurité crée une relation étroite entre le serveur et un adaptateur. Le protocole SSL est une forme de sécurité symétrique.

De nombreux logiciels de sécurité tiers sont basés sur le protocole **Secure Sockets Layer** (SSL - sécurité à couche de connexion sécurisée). Le protocole SSL fournit le chiffrement de données, l'authentification du serveur, l'intégrité des messages et

une authentification facultative du client pour les connexions TCP/IP. Le protocole SSL est habituellement étroitement associé à des systèmes qui utilisent la sécurité symétrique, toutes les données sont donc chiffrées, qu'elles soient sensibles ou non. Le protocole SSL ne peut pas reconnaître des données sensibles d'autres qui ne le sont pas. Lorsque les niveaux de sécurité entre les messages entrants et sortants peuvent être différents, il s'agit de sécurité asymétrique. Par exemple, supposons qu'un utilisateur ait besoin de s'authentifier auprès d'un serveur, à l'aide d'un identificateur de l'utilisateur et d'un mot de passe, avant de recevoir un document volumineux en provenance du serveur. L'identificateur de l'utilisateur et le mot de passe doivent être chiffrés, afin de ne pas être interceptés pendant le transfert par un utilisateur non autorisé, tandis que le document n'a pas besoin d'être chiffré pendant la transmission. Lorsque le protocole SSL est utilisé, le chiffrement concerne à la fois l'identificateur de l'utilisateur, le mot de passe et le document. Le chiffrement du document volumineux peut avoir un impact négatif sur les performances du serveur.

Avec la sécurité asymétrique, le flux de messages en provenance de l'utilisateur peut être défini pour chiffrer l'identificateur de l'utilisateur et son mot de passe, tandis que le flux en provenance du serveur peut être défini de façon à ne pas être chiffré. L'ID utilisateur et son mot de passe sont protégés par chiffrement, tandis que les performances du serveur sont optimisées en évitant de chiffrer le document volumineux avant de l'envoyer à l'utilisateur. La sécurité asymétrique permet également de mettre à jour dynamiquement les paramètres de sécurité entre les parties communiquant entre elles. Avec la sécurité asymétrique, tel le protocole SSL, les paramètres de sécurité sont établis avant le début de la communication. Les modifications des paramètres de sécurité ne prennent pas effet avant le début d'une nouvelle session de communication. Avec la sécurité asymétrique, les modifications des paramètres de sécurité peuvent être dynamiques car les parties communiquant entre elles ont des définitions individuelles de leurs paramètres de sécurité. La modification des paramètres de sécurité de l'une des parties n'interrompt pas la session de sécurité en cours et n'exige pas la mise à jour de l'autre partie. La sécurité asymétrique peut procurer plus de flexibilité mais sa faiblesse est qu'elle repose directement sur les noeuds finaux laissant les noeuds intermédiaires vulnérables, notamment aux manipulations de données non autorisées.

Avec la confidentialité complète, la sécurité peut s'établir au niveau des applications pour sécuriser les messages avant qu'ils entrent dans une application ou en sortent. Ce type de sécurité élimine le risque de voir des sources non autorisées accéder aux messages pendant qu'ils sont sur les noeuds intermédiaires. La confidentialité complète permet aussi d'associer aux différents types de message des niveaux de sécurité appropriés.

Les niveaux de sécurité

Le niveau de sécurité est configuré pour chaque type de message, pour les adaptateurs individuels et InterChange Server. Il existe quatre niveaux de sécurité :

Néant Aucun niveau de sécurité n'est fixé, les messages sont donc envoyés de façon normale depuis l'adaptateur ou InterChange Server. Il s'agit du niveau de sécurité par défaut.

Intégrité

L'expéditeur, soit un adaptateur, soit InterChange Server, ajoute une signature numérique au message. Le réceptionnaire vérifie la signature à l'aide de la clé publique de l'expéditeur. Le réceptionnaire authentifie l'expéditeur en vérifiant la signature, car seul l'expéditeur a accès à sa propre clé privée, et a donc pu l'utiliser pour générer la signature. La

signature est une synthèse de message des données, que le réceptionnaire peut créer pour vérifier si elle correspond à celle qu'il a reçue, auquel cas l'intégrité du message est garantie. L'intégrité implique également l'authentification, car lors du processus de vérification que les données n'ont pas été modifiées, l'identité de l'expéditeur est contrôlée.

Confidentialité

L'expéditeur chiffre complètement le message. Le réceptionnaire déchiffre le message et le transmet pour la suite du traitement. Lorsqu'un adaptateur ou InterChange Server envoie un message en mode confidentialité, il génère une clé secrète ; chiffre le message avec la clé secrète ; chiffre la clé secrète avec la clé publique du réceptionnaire ; joint la clé secrète chiffrée au message ; puis il envoie le message. Le réceptionnaire déchiffre la clé secrète avec sa propre clé privée ; déchiffre le message avec la clé secrète déchiffrée ; puis transmet le message pour la suite du traitement.

La clé secrète est chiffrée avec la clé publique du réceptionnaire. La clé secrète chiffrée ne peut être déchiffrée que par la clé privée du réceptionnaire, la sécurité du message s'en trouve ainsi garantie, puisque seul le réceptionnaire a accès à sa propre clé privée.

Intégrité plus confidentialité

L'expéditeur ajoute une signature numérique à chaque message, comme cela est décrit dans la partie Intégrité, et chiffre complètement le message, selon le processus décrit dans la partie Confidentialité.

InterChange Server et les adaptateurs gèrent leurs propres magasins de clés. Un magasin de clés est un fichier protégé par un mot de passe utilisé pour stocker de façon sûre les clés publiques et privées. InterChange Server gère un magasin de clés qui contient ses propres clés publiques et privées ainsi que les clés publiques de tous les adaptateurs installés. Un adaptateur individuel gère un magasin de clés qui contient ses propres paires de clés publiques et privées, la clé publique d'InterChange Server ainsi que les clés publiques de tous les processus avec lesquels il communique.

Les combinaisons des niveaux de sécurité et des types de messages à sécuriser sont possibles, à la fois pour les adaptateurs individuels et pour InterChange Server. Par exemple, l'intégrité peut être configurée pour les messages administratifs en provenance d'InterChange Server, tandis que la confidentialité est réservée aux objets métier, et aucune sécurité n'est définie pour l'adaptateur de courriel.

Les types de messages

Il existe quatre types de messages, en provenance d'un adaptateur ou d'InterChange Server, pouvant être sécurisés :

Tous les messages (All)

Tous les messages en provenance d'un adaptateur ou d'InterChange Server sont sécurisés.

Messages administratifs (Admin)

Tous les messages administratifs en provenance d'un adaptateur ou d'InterChange Server sont sécurisés.

Tous les objets métier (BO)

Tous les objets métier en provenance d'un adaptateur ou d'InterChange Server sont sécurisés.

Objets métier individuels (BO nom de la spécification)

Des messages d'objets métier spécifiques en provenance d'un adaptateur ou d'InterChange Server sont sécurisés en précisant le nom de la spécification.

Pour sécuriser les messages avec une confidentialité complète, l'administrateur doit déterminer le type de documents à sécuriser et le niveau de sécurité à appliquer, puis configurer à la fois InterChange Server et les adaptateurs individuels. Lorsque la confidentialité complète est configurée pour InterChange Server, elle ne s'applique qu'aux messages envoyés à partir d'InterChange Server. Lorsque la confidentialité complète est configurée pour un adaptateur individuel, elle ne s'applique qu'aux messages envoyés à partir de cet adaptateur individuel. Tous les événements existants doivent être supprimés du système avant de passer en mode confidentialité complète. Les événements existants ne seront pas traités correctement lors de la configuration en confidentialité complète. Pour des informations complémentaires sur la configuration de la confidentialité complète, reportez-vous au manuel *System Administration Guide*.

Le contrôle d'accès basé sur les rôles

Pour empêcher les utilisateurs non autorisés d'accéder à quelque système que ce soit, les ID utilisateurs et les mot de passe représentent la sécurité de base. Un ID utilisateur est délivré à un utilisateur, qui peut être un individu ou un processus qui souhaite accéder au système. Le fait d'authentifier un utilisateur ne veut pas dire qu'il peut accéder à toutes les parties du système. Il peut y avoir des parties du système auxquelles seuls des utilisateurs sélectionnés doivent pouvoir accéder, tandis que d'autres parties du système sont accessibles à tous les utilisateurs. Le contrôle d'accès fait référence aux restrictions d'accès à des parties du système en fonction de l'identité de l'utilisateur et des droits qui lui ont été accordés. Des utilisateurs individuels peuvent se voir accorder des autorisations d'accès aux différents composants, mais la gestion des autorisations peut dans ce cas prendre beaucoup de temps aux administrateurs. Le contrôle des accès basé sur les rôles authentifie les utilisateurs qui veulent accéder à InterChange Server, puis limite l'accès des utilisateurs à certains composants individuels, en fonction du rôle qui a été attribué à l'utilisateur.

Rôles

Un rôle est un groupe d'un ou plusieurs utilisateurs qui forment une association fonctionnelle. L'attribution d'autorisations en fonction des rôles diminue la charge administrative de façon significative. Pour chaque opération nécessitant une sécurisation, l'administrateur définit les rôle qui doivent être accordés pour réaliser l'opération. Seuls les utilisateurs qui sont membres du rôle autorisé peuvent participer à l'opération. Lorsque des autorisations sont attribuées aux rôles, les autorisations du rôles représentent la stratégie de sécurité du rôle. Les utilisateurs peuvent se voir attribuer de multiples rôles.

Une opération ou une action n'est pas sécurisée si aucun rôle ne lui est attribué, et tous les utilisateurs authentifiés peuvent y participer. Les utilisateurs ne peuvent effectuer une action sécurisée que s'ils sont membres du rôle qui dispose des droits d'accès à l'action.

Un rôle prédéfini d'administrateur a été inclus à InterChange Server en tant que rôle par défaut. Les membres du rôle administrateur ont l'autorisation d'effectuer toutes les opérations sur le serveur, y compris de créer des utilisateurs et des rôles, ainsi que d'accorder des autorisations aux rôles. L'administrateur a la possibilité de créer des rôle supplémentaires qui correspondent davantage à la structure de l'entreprise. Par exemple, une entreprise peut disposer de plusieurs rôles de

développeurs, tels que DéveloppeurFinance ou DéveloppeurRH, pour restreindre l'accès aux composants développés par les différents groupes. Les utilisateurs peuvent appartenir à plusieurs rôles. Les rôles prédéfinis ne peuvent pas être supprimés.

Un administrateur peut interroger le système pour savoir quels utilisateurs sont connectés au système. Un utilisateur peut se connecter à un maximum de 20 sessions à la fois. L'administrateur peut déconnecter une session particulière de n'importe quel utilisateur connecté. L'administrateur peut également déconnecter toutes les sessions ouvertes d'un utilisateur, ce qui permet à l'administrateur de déconnecter un utilisateur qui a oublié de se déconnecter. Une erreur est signalée lorsqu'un utilisateur essaie de se connecter alors qu'il a déjà 20 sessions ouvertes. L'administrateur peut également réinitialiser le mot de passe d'un utilisateur.

Un utilisateur invité a été prédéfini en tant qu'utilisateur par défaut, et se voit attribuer un mot de passe d'invité. Les utilisateurs par défaut ne peuvent pas être supprimés. Le nombre de connexions avec le rôle d'invité est illimité.

L'administrateur InterChange Server peut importer et exporter des stratégies de sécurité, qui représentent les autorisations attribuées vis-à-vis des composants, via le fichier `security.xml`. Les définitions et les informations relatives aux rôles peuvent être exportées et importées avec l'outil de ligne de commande `repos_copy` par un utilisateur disposant du rôle approprié. Le fichier s'appelle `RoleMembership.xml`. Les utilisateurs et les mots de passe peuvent être exportés vers un fichier binaire et importés de ce fichier binaire par un autre serveur. L'exportation des mots de passe représente un risque pour la sécurité. Ce fichier doit être exporté vers un répertoire qui est sécurisé par le système d'exploitation et doit être supprimé après l'importation.

La stratégie de sécurité par défaut veut que seul un utilisateur ayant le rôle d'administrateur peut démarrer, arrêter, gérer la sécurité, exporter et déployer des fichiers de configuration dans InterChange Server. Les autorisations par défaut d'un nouveau composant permettent à un utilisateur authentifié d'effectuer tous types d'actions. Un composant n'est pas sécurisé par défaut.

Le contrôle des accès basé sur les rôles n'est *pas* configuré automatiquement lors de l'installation d'InterChange Server. Pour configurer le contrôle des accès basé sur les rôles, référez-vous au manuel *System Administration Guide*. Avant d'activer le contrôle des accès basé sur les rôles, il est nécessaire d'ajouter au moins un utilisateur avec le rôle d'administrateur. Si InterChange Server utilise une implémentation à référentiel, il faut créer un utilisateur pour l'ajouter au rôle d'administrateur. Si le contrôle des accès basé sur les rôles a été activé et qu'il n'y a pas au moins un utilisateur avec le rôle d'administrateur, le serveur émet un message d'erreur et démarre avec le contrôle des accès basé sur les rôles désactivé. Aucun ID utilisateur ni mot de passe n'est alors exigé pour démarrer le serveur, et tous les utilisateurs ont accès à toutes les opérations. Lorsque le contrôle des accès basé sur les rôles est activé, un ID utilisateur et un mot de passe sont exigés pour démarrer le serveur, et tous les contrôles d'autorisations ou d'accès sont opérationnels.

Contrôle

Le contrôle permet aux administrateurs d'effectuer le suivi des utilisateurs ayant réalisé des opérations sécurisées, telles les connexions, les déconnexions, la gestion des utilisateurs, des rôles et des autorisations. Les données du contrôle qui sont collectées incluent la date, l'heure, l'ID utilisateur, l'opération réalisée et le statut de l'opération (succès, accès refusé ou erreur). Le journal de suivi n'est pas écrit

dans le fichier habituel `Interchangesystem.log`. Il est écrit dans un fichier séparé dans le répertoire `your_ics_dir_name/log/audit directory`, qui n'est pas sécurisé. Le fichier du journal de suivi contient des informations qui devraient être sécurisées par le système d'exploitation, afin d'empêcher les utilisateurs non autorisés d'y accéder. Il est possible de paramétrer la taille du fichier journal et la fréquence avec laquelle de nouveaux fichiers journaux sont créés. Le nom du fichier dépend de la fréquence avec laquelle de nouveaux fichiers journaux doivent être créés, une fois que le fichier journal a atteint la taille maximale définie dans les paramètres. Les options de fréquence sont : quotidien, hebdomadaire et mensuel. La valeur par défaut est hebdomadaire. Le format de fichier du journal de suivi est le suivant :

```
Fichier quotidien : csaudit_yyyymmdd_nnn.log
Fichier hebdomadaire : icsaudit_yyyymmwx_nnn.log
    Où x = 1,2,3... Numéro de semaine
Fichier mensuel : icsaudit_yyyymm_nnn.log
```

où `yyyy` est l'année, `mmm` est le mois, `dd` est le jour, `wx` est le numéro de semaine et `nnn` = 001.....999, est le numéro de fichier lorsque plusieurs fichiers de suivi doivent être créés une fois la taille maximale atteinte. Les opérations pouvant être contrôlées sont les suivantes :

- Connexion et déconnexion
- Serveur : démarrage et arrêt, sécurité, gestion des rôles et des utilisateurs, surveillance, affichage des erreurs. MMS : déploiement, exportation, suppression, compilation, exportation et déploiement des fichiers de configuration
- Objet de collaboration : démarrage, arrêt, pause, arrêt du système, exécution (appel d'accès à la structure), résolution des états transactionnels, soumission des événements ayant échoué, suppression des événements ayant échoué, annulation d'un flux processus métier à longue échéance
- Connecteur : démarrage, arrêt, pause, arrêt de l'agent, soumission des événements ayant échoué, suppression des événements ayant échoué
- Mappe : démarrage et arrêt
- Relation : démarrage et arrêt
- Test de performances : démarrage et arrêt

Registre d'utilisateurs

Le registre d'utilisateurs est le système qui stocke les noms des utilisateurs et les mots de passe. Par défaut, les utilisateurs et les mots de passe sont stockés dans le référentiel WBI local. Le paramètre de configuration `userRegistry` peut être utilisé pour configurer InterChange Server afin qu'il utilise un annuaire LDAP en tant que registre d'utilisateurs. Le protocole LDAP est un protocole permettant de gérer l'accès aux services d'annuaire des entreprises. Les annuaires d'entreprises sont utilisés pour stocker les informations pouvant être partagées par plusieurs applications. Par exemple, parmi les informations utilisateur pouvant être nécessaires à toutes les applications de l'entreprise, il pourrait y avoir le nom d'utilisateur, le mot de passe et l'adresse de courriel. Si ces données sont stockées de façon individuelle dans la base de données propriétaire de chaque application, les données sont dupliquées, ce qui en complique la synchronisation. Lorsqu'un nouvel employé rejoint la société, un ID utilisateur et un mot de passe doivent alors être créés dans chaque application pour ce nouvel employé. En revanche, avec un service d'annuaire, l'ID utilisateur et le mot de passe ne doivent être créés qu'une fois dans l'annuaire, et les applications de l'entreprise peuvent ainsi réutiliser ces informations.

Le fournisseur est défini sur LDAP, afin d'utiliser un annuaire LDAP pour le référentiel utilisateur, tandis qu'il est défini sur REPOS pour une utilisation du référentiel WBI local en tant que registre d'utilisateurs. Lorsque le référentiel WBI

est utilisé pour stocker les utilisateurs et les mots de passe, il est fortement conseillé d'utiliser une base de données séparée pour cela. L'URL de la base de données qui héberge le registre d'utilisateurs peut être spécifiée. Il s'agit d'une propriété hiérarchique appelée USER_REGISTRY dans le fichier de configuration interchangesystem.cfg. La structure est identique à celle de la propriété du référentiel existant, dans le sens qu'une URL de la base de données, un nom d'utilisateur et un mot de passe peuvent être spécifiés. La propriété peut être éditée à l'aide de l'outil de configuration du serveur. Il est recommandé de ne mettre en oeuvre qu'un seul registre d'utilisateurs par entreprise, que ce soit en utilisant le protocole LDAP ou une base de données WBI, afin de permettre à plusieurs instances d'InterChange Server de partager le même registre d'utilisateurs sans avoir à importer et à exporter les utilisateurs et les mots de passe. InterChange Server peut utiliser le nom d'utilisateur et le mot de passe à partir d'un annuaire LDAP, en utilisant le schéma InetOrgPerson selon la description du document RFC 2798. Le schéma InetOrgPerson a l'identificateur d'objet (OID) 2.16.840.1.113730.3.2.2, qui sont des numéros affectés par le groupe de travail IETF, qui donne un identificateur unique à chaque schéma. InterChange Server n'impose pas de règle pour l'arborescence à l'intérieur de l'annuaire. Les utilisateurs peuvent spécifier un nom spécifique de base (baseDN) formant partie de la configuration des utilisateurs et des rôles. Ce nom spécifique de base (baseDN) sert de racine à toutes les recherches et les mises à jour.

La sécurisation des composants et des actions

La liste des composants pouvant être sécurisés, en plus des actions sécurisables, inclut :

Tableau 2. Composants sécurisables

Composants sécurisables	Actions sécurisables
InterChange Server	<ul style="list-style-type: none"> • Démarrage • Arrêt du système • Sécurité - gestion des rôles et des utilisateurs • Surveillance • Affichage des événements ayant échoué • MMS - Déployer • MMS - Exporter • MMS - Supprimer • MMS - Compiler • MMS - Exporter les fichiers de configuration • MMS - Déployer les fichiers de configuration
Objets de collaboration	<ul style="list-style-type: none"> • Démarrer • Arrêter • Faire une pause • Arrêter le système • Exécuter (appel à la structure d'accès) • Résoudre l'état transactionnel • Transmettre les événements ayant échoué • Supprimer les événements ayant échoué • Annuler le flux LLBP

Tableau 2. Composants sécurisables (suite)

Composants sécurisables	Actions sécurisables
Connecteurs	<ul style="list-style-type: none"> • Démarrer • Arrêter • Faire une pause • Quitter l'agent • Transmettre les flux ayant échoué • Supprimer les flux ayant échoué
Mappes	<ul style="list-style-type: none"> • Démarrer • Arrêter
Relations	<ul style="list-style-type: none"> • Démarrer • Arrêter
Tests de performances	<ul style="list-style-type: none"> • Démarrer • Arrêter

Récapitulatif

Ce chapitre vous a présenté les principaux composants du système IBM WebSphere InterChange Server. Les principaux points à retenir sont :

- Les collaborations interagissent avec des connecteurs et avec l'interface d'accès au serveur pour créer des processus métier répartis. Les collaborations contiennent la logique des processus métier ; les collaborations construisent un pont entre le système IBM WebSphere InterChange Server et l'environnement applicatif pour faciliter l'implémentation de la logique des processus métier dans les applications. L'interface d'accès au serveur et le connecteur XML vous permettent d'étendre la logique des processus métier via Internet.
- Toutes les interactions impliquant une collaboration s'effectuent par échange d'objets métier. Lorsque l'interface d'accès au serveur reçoit un appel, ou lorsqu'un connecteur reçoit des informations à propos d'un événement d'application, un objet métier est créé et envoyé à la collaboration.
- Un objet métier est générique ou propre à l'application. Les collaborations traitent des objets métier génériques, et les connecteurs traitent des objets métier propres à une application.
- InterChange Server est le centre d'exécution des collaborations. Il comprend un référentiel, des contrôleurs de connecteur, le service de gestion des événements, le service de connectivité des bases de données, et le service Transactions.
- System Manager affiche, configure, et contrôle l'exécution des objets stockés sur chaque serveur InterChange Server.
- Les composants sont répartis. La collaboration/le protocole du connecteur, la messagerie, et la fonction ORB représentent le ciment entre les différents composants.
- Les messages qui circulent via InterChange Server peuvent être sécurisés grâce à la confidentialité complète. InterChange Server peut être sécurisé à l'aide du contrôle des accès basé sur les rôles.

Le chapitre suivant étudie plus en détail les collaborations.

Chapitre 2. Outils à utiliser avec InterChange Server

Les outils suivants sont disponibles pour une utilisation avec InterChange Server :

- «Boîte à outils de WebSphere Business Integration»
- «Outils de développement» à la page 45
- «Outils d'administration» à la page 46

Boîte à outils de WebSphere Business Integration

La boîte à outils de WebSphere Business Integration fournit des outils d'administration et de développement utilisables avec InterChange Server. La boîte à outils de WebSphere Business Integration est prise en charge par Windows 2000 et Windows XP. Les outils d'administration comprennent :

- System Manager
- System Monitor
- Flow Manager
- Log Viewer
- Relationship Manager

Pour des informations complémentaires sur les outils et les tâches administratives, reportez-vous au manuel *System Administration Guide*.

Les outils de développement comprennent :

- Map Designer
- Relationship Manager
- Process Designer
- Business Object Designer
- Connector Configurator

Ces outils de développement sont décrits plus en détail dans le manuel *Implementation Guide for WebSphere InterChange Server*, et les guides de développement des différents composants.

Vous pouvez accéder aux outils de la boîte à outils de WebSphere Business Integration par l'intermédiaire d'un raccourci Windows en sélectionnant **Démarrer > Programmes > WebSphere Business Integration Toolset**, puis en sélectionnant **Administratif** pour les outils administratifs, ou **Développement** pour les outils de développement.

La boîte à outils de WebSphere Business Integration fournit les droits pour une utilisation avec IBM Tivoli License Manager (ITLM). Il fournit une prise en charge pour des fonctions d'inventaire. ITLM offre un meilleur contrôle et une plus grande souplesse sur les sommes versées pour les logiciels, et permet aux clients de payer en fonction de l'utilisation réelle plutôt qu'en fonction d'une utilisation totale sur une machine. ITLM mesure l'utilisation des produits et fournit le compte-rendu correspondant à IBM. Les fonctions comprennent :

- La récupération des données d'inventaire, ainsi que l'extraction des informations sur les produits utilisés et installés sur chaque site.

- Stockage et maintenance des informations sur les fournitures de logiciels et les contrats des produits achetés par les différents clients ou organismes indépendants, ainsi que les modalités des licences.
- Comparaison entre les licences fournies, installées et utilisées, et synchronisation des informations obtenues d'après ces trois vues différentes.
- Compilation des décomptes et des résultats de l'inventaire pour identifier les droits de licences nécessaires pour chaque produit et fourniture du nombre nécessaire de licences.
- Vérification continue que les logiciels sont utilisés conformément aux accords existants entre les clients et les fournisseurs.

System Manager

L'interface de System manager provient de la plate-forme Eclipse — un environnement de développement open-source intégré de création d'outils. La plate-forme Eclipse fournit aux développeurs d'outils un kit de développement et un contexte d'exécution qui permettent au développeur d'écrire des plugiciels pour autoriser l'utilisateur à travailler avec un type de ressource particulier.

Remarque : System Manager est installé en tant que plugiciel pouvant être utilisé dans les versions badgées IBM de la plate-forme Eclipse.

Vous accédez à System Manager avec une *perspective* basée sur Eclipse. La perspective System Manager fournit des vues et des éditeurs qui vous aident d'une façon générale dans l'utilisation de votre système et l'accès aux outils, ainsi qu'en vous fournissant une interface pour déployer les projets d'intégration que vous développez. Ces perspectives comprennent la prise en charge des tâches suivantes :

- Accès aux outils de développement d'IBM WebSphere InterChange Server
- Réalisation des tâches de configuration sur les composants modulaires individuels utilisés dans un système IBM WebSphere InterChange Server
- Travail avec des instances d'InterChange Server
- Gestion des groupes de composants d'intégration, comme les projets utilisateurs, et déploiement des composants vers le référentiel
- Étalonnage des caractéristiques de performances d'un système d'intégration de processus d'entreprise et de ses composants

Dans System Manager, les tâches sont accessibles par un menu et la barre d'outils correspondante, ainsi que par la vue du System Manager de WebSphere Business Integration. La vue de WebSphere Business Integration System Manager est affichée dans le volet gauche de System Manager, elle représente une liste de type dossier des bibliothèques de composants d'intégration modulaires (tels que les définitions d'objets métier, les pools de connexions à une base de données et les objets et modèles de collaboration). Vous pouvez réaliser des tâches de développement et de configuration en cliquant avec le bouton droit sur l'icône d'une bibliothèque pour accéder à ses composants.

À partir de System Manager, vous pouvez accéder aux outils de l'interface graphique, pour créer des modèles de collaborations, des définitions d'objets métier, des mappes et des relations. Notez bien que vous pouvez également accéder à certains outils de développement et les exécuter indépendamment de System Manager.

Consultez «Outils de développement» à la page 45 pour obtenir une liste partielle des outils de développement disponibles.

Configuration des composants

Vous pouvez utiliser System Manager pour manipuler les composants modulaires d'IBM WebSphere InterChange Server, appelés **composants d'intégration**, qui sont utilisés pour implémenter les échanges de données de gestion dans un environnement IBM WebSphere InterChange Server. Ces composants comprennent les : collaborations, connecteurs, objets métier, mappes, relations et autres pools de connexions à une base de données. À partir de System Manager, vous pouvez réaliser des tâches de configuration à base de composants telles que :

- La configuration d'objets de collaboration à partir de modèles de collaboration, y compris la liaison de leurs ports et les propriétés générales avec lesquelles ils s'exécutent dans InterChange Server.
- L'accès aux outils d'affichage et de révision des définitions du référentiel des propriétés des connecteurs standards et spécifiques, et la définition des objets métier et mappes utilisés par les connecteurs.

Modes de System Manager et d'InterChange Server

Lorsque vous développez des composants d'intégration, vous pouvez sélectionner un mode de serveur dans System Manager, pour contrôler la façon et l'étendue selon lesquelles les définitions et les configurations des composants que vous avez créés seront chargés dans le référentiel. Les sélections des modes serveurs sont disponibles pour les différentes étapes du développement : conception, test et production. Pour des informations complémentaires sur l'utilisation des modes du serveur, consultez le manuel *Implementation Guide for WebSphere InterChange Server*.

Outils de développement

Vous pouvez créer et modifier des collaborations, des connecteurs, des objets métier, des mappes et des relations. Les outils de développement sont utilisés pour créer et configurer les composants du système dans un environnement de développement, avant de mettre le système en production. Ces outils sont décrits plus en détail dans le manuel *Implementation Guide for WebSphere InterChange Server*. Le tableau 3 liste et décrit une partie des outils logiciels disponibles dans un environnement de développement.

Tableau 3. Outils de développement

Cible	Outil	Description
Développeurs de collaborations	Process Designer	Outil graphique avec lequel vous pouvez définir des modèles de collaborations. Pour des informations complémentaires sur le concepteur de processus et sur la manière de développer des collaborations, consultez le manuel <i>Collaboration Development Guide</i> .
Les développeurs de collaborations et de mappes	Environnement de test et Virtual Test Connector	Fournit un environnement dans lequel vous pouvez tester les interfaces d'intégration de processus d'entreprise que vous avez développées. Fournit des interfaces graphiques pour émuler les connecteurs, démarrer les composants requis et examiner les données de l'objet métier. Pour des informations complémentaires sur les environnements de test et la façon de tester les collaborations, les mappes et les connecteurs, consultez le manuel <i>Implementation Guide for WebSphere InterChange Server</i> .

Tableau 3. Outils de développement (suite)

Cible	Outil	Description
Développeurs de connecteurs	Connector Configurator	Utilisé pour ajouter des propriétés propres à l'application à une définition de connecteur, configurer les valeurs des propriétés, et configurer la définition du connecteur avec ses objets métier et ses mappes. Pour plus d'information sur Connector Configurator et la façon de développer des connecteurs, consultez les manuels suivants : <i>Connector Development Guide for Java</i> ou <i>Connector Development Guide for C++</i> .
Développeurs de mappes	Map Designer	Un outil graphique qui définit les transformations de données. Pour des informations complémentaires sur Map Designer et la façon de développer les mappes, consultez le manuel <i>Map Development Guide</i> .
Développeurs de mappes	Relationship Designer	Un outil graphique qui définit les relations entre les types d'objets. Ces relations sont importantes pour le mappage, pour indiquer, par exemple, la relation entre un type d'objet métier et un autre. Pour des informations complémentaires sur Relationship Designer et sur la manière de développer des relations, consultez le manuel <i>Map Development Guide</i> .
Développeurs d'objets métier	Business Object Designer	Une interface à base de formulaires pour créer des définitions d'objets métier aussi bien manuellement qu'à partir d'agents de détection d'objets (ODA). Pour des informations complémentaires sur Business Object Designer et la façon de concevoir des définitions d'objets métier, consultez le manuel <i>Business Object Development Guide</i> .
Développeurs d'objets métier	Object Discovery Agent Development Kit (ODK)	Une interface de programme d'application (API) qui facilite la création d'agents de détection d'objets (ODA). Les ODA identifient la configuration des objets métier requise spécifiquement pour la source de données afin de générer des définitions à partir de ces spécifications. Pour des informations complémentaires sur Business Object Designer et la façon de développer des agents de détection d'objets (ODA), consultez le manuel <i>Business Object Development Guide</i> .
Tous développeurs	Assistant d'étalonnage	L'outil d'étalonnage vous permet de tester différents composants, interfaces, et systèmes d'IBM WebSphere, afin de mesurer leurs performances. Vous accédez à l'outil d'étalonnage par System Manager. Pour des informations complémentaires sur l'assistant d'étalonnage, consultez le manuel <i>Benchmarking Guide</i> .

Outils d'administration

System Manager fournit un accès à certains outils d'administration, y compris à System Monitor pour gérer les états des composants d'intégration. Pour des informations complémentaires sur les outils d'administration et les tâches administratives, reportez-vous au manuel *System Administration Guide*.

Chapitre 3. Collaborations

Dans l'implémentation d'InterChange Server, le terme **collaborations** désigne des modules logiciels qui contiennent du code et une logique de processus métier qui gère les interactions entre les applications. Une collaboration peut être simple, et ne comporter que quelques étapes, ou complexe, et comprendre plusieurs étapes et d'autres collaborations.

Les collaborations peuvent être réparties entre un nombre illimité d'applications, et prendre en charge des interventions sur appels synchrones et asynchrones, ainsi que des processus métier de longue durée. Ce chapitre décrit les collaborations, ce en quoi elles consistent, et comment vous interagissez avec elles.

Ce chapitre présente les collaborations et les composants que vous utilisez pour voir, modifier et créer les collaborations. Les sections suivantes seront abordées :

- «Objets et les modèles de collaborations»
- «Traitement de la collaboration» à la page 48
- «Collaborations et les traitements simultanés» à la page 49
- «Groupes de collaborations» à la page 50
- «Ports» à la page 50
- «Scénarios» à la page 53
- «Logique du processus métier» à la page 54
- «Interactions avec les connecteurs et les applications» à la page 56
- «Initialisation de la collaboration» à la page 58
- «Récapitulatif» à la page 58

Pour obtenir des instructions détaillées sur l'utilisation de l'outil de collaboration d'IBM WebSphere InterChange Server pour créer des modèles, référez-vous au *Collaboration Development Guide*.

Objets et les modèles de collaborations

Lorsque vous installez une collaboration, vous installez un **modèle de collaboration**. Un modèle de collaboration contient l'intégralité de la logique d'exécution de la collaboration, mais ce n'est pas un exécutable. Pour exécuter une collaboration, vous devez tout d'abord créer un **objet de collaboration** à partir du modèle. L'objet de collaboration devient exécutable une fois que vous avez configuré sa liaison avec des connecteurs ou d'autres objets de collaboration, et en spécifiant d'autres propriétés de configuration.

Remarque : La boîte à outils fournie avec InterChange Server comprend un outil de conception de collaborations appelé **Process Designer**, avec lequel vous pouvez personnaliser et créer des modèles de collaboration.

Vous pouvez utiliser les mêmes modèles de collaboration pour créer de multiples objets de collaboration. Par exemple, supposons que vous utilisiez une collaboration Contact Manager pour intégrer des applications de guichet et d'arrière guichet. Si l'un des services de la société utilise des applications de guichet Clarify et qu'un autre service utilise Siebel, alors que ces deux services

utilisent des applications d'arrière-guichet SAP, vous pouvez créer deux objets de collaboration à partir du modèle ContactManager.

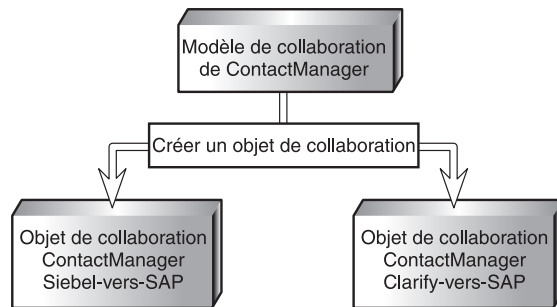


Figure 17. Création d'objets de collaboration à partir d'un modèle

Vous pouvez alors configurer chaque objet de collaboration de sorte que chacun des deux soit relié à des connecteurs différents pour leur application source mais qu'ils soient tous les deux reliés au même connecteur pour l'application de destination.

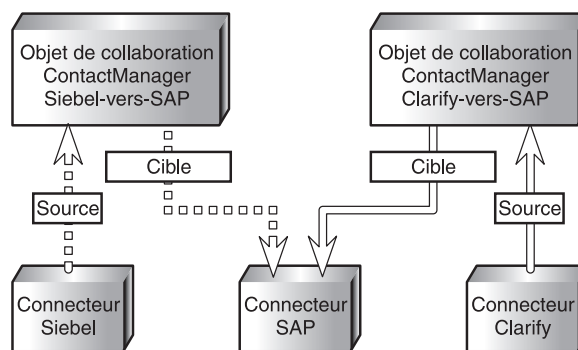


Figure 18. Configuration des objets de collaboration

Remarque : Ce manuel utilise le terme "collaboration" pour désigner le comportement, la conception et les fonctions. Les termes "modèle de collaboration" et "objet de collaboration" ne sont utilisés que lorsqu'il est important de les distinguer, en général lorsqu'il est question du processus de configuration ou du référentiel.

Traitement de la collaboration

Une collaboration initie un flux du procédé lorsqu'elle en reçoit l'ordre, qui est déclenché par l'arrivée d'un objet métier. Le déclencheur peut être l'un des éléments suivants :

- Un objet métier provenant d'un événement publié par un connecteur lors d'une interaction publication-abonnement (décrite dans le chapitre 1)
- Une requête d'accès reçue par l'intermédiaire d'une interface d'accès au serveur (décrite dans le chapitre 1)
- Une intervention sur appel d'un connecteur, décrite ci-après dans ce chapitre

Gestion des interventions sur appel et les processus métier longue durée

Les collaborations peuvent interagir avec les connecteurs tout en conservant le contexte de flux de données de l'intervention sur appel pendant une période déterminée, sans obliger le processus d'intervention sur appel à attendre de façon synchrone. Si une réponse appropriée à l'intervention sur appel est reçue dans la période spécifiée, le flux de données reprend et le traitement continue. Cette fonction est appelée **processus métier longue durée** pour les collaborations, et peut être activée lors de la création du modèle de collaboration.

Les collaborations d'une implémentation InterChange Server peuvent prendre en charge les types suivants d'interventions sur appel :

- Communication synchrone sortante de la collaboration
- Communication asynchrone sortante de la collaboration
- Communication asynchrone entrante vers la collaboration

Intervention sur appel synchrone sortant

L'intervention sur appel synchrone sortant utilise un mécanisme synchrone de requête/réponse. L'intervention sur appel envoie la requête mais reste en attente jusqu'à ce que la réponse arrive et soit traitée.

Les interventions sur appel synchrones prennent en charge la compensation. De plus, elles supportent une valeur de délai d'attente pour les processus métier longue durée.

Intervention sur appel asynchrone sortant

L'intervention sur appel asynchrone sortant envoie une requête à partir de la collaboration mais n'attend pas de réponse pour continuer son traitement.

Les interventions sur appel asynchrones prennent en charge la compensation, mais elles ne supportent pas de valeur de délai d'attente pour les processus métier longue durée.

Intervention sur appel asynchrone entrant

Une intervention sur appel asynchrone entrant attend la réception d'un événement entrant et est utilisée en conjonction avec des processus métier longue durée. Lorsqu'une intervention sur appel asynchrone entrant est créée, une valeur de délai d'attente lui est donnée ; si l'intervention sur appel ne reçoit pas d'événement entrant avant l'expiration du délai d'attente, cela provoque une exception.

Les interventions sur appel asynchrone entrant ne sont disponibles que si l'option processus métier longue durée du modèle de collaboration a été activée. Cette fonction n'est utilisée que pour les échanges avec les connecteurs.

Les interventions sur appel asynchrones ne prennent pas en charge la compensation.

Collaborations et les traitements simultanés

Une collaboration initie un flux du procédé lorsqu'elle en reçoit l'ordre, qui est déclenché par l'arrivée d'un objet métier. Par défaut, la collaboration termine le traitement d'un flux avant de traiter le flux initié par le déclenchement de l'objet métier suivant.

Cependant, vous pouvez configurer une collaboration pour qu'elle traite simultanément plusieurs flux déclenchés par des événements. Si une collaboration est configurée de cette façon, elle va identifier tous les flux qui ont des interdépendances d'objets métier pour les traiter dans l'ordre, mais elle pourra traiter simultanément les flux qui n'ont pas de conflit.

Groupes de collaborations

Un objet de collaboration peut être lié à un autre objet de collaboration pour former un **groupe de collaborations**. Un groupe de collaborations utilise la puissance de plusieurs collaborations discrètes pour intégrer les processus métier correspondants.

Dans un groupe de collaborations, les relations entre les collaborations ne sont limitées que par le fait qu'une collaboration fournit des objets métier à une autre collaboration. Les groupes de collaborations peuvent être connectés selon le nombre souhaité de topologies, telles que des chaînes ou même des webs.

Les groupes de collaborations sont également utiles pour l'isolement des données. Pour des détails complémentaires, consultez le *Collaboration Development Guide*.

Ports

Une collaboration dispose d'un ensemble d'interfaces avec le monde extérieur appelées **ports**. Dans un modèle de collaboration, chaque port est une variable qui représente un objet métier que l'objet de collaboration reçoit ou produit en contexte d'exécution. Par exemple, imaginez une collaboration d'un hypothétique service de facturation qui est déclenchée par la fermeture d'un dossier dans une application de gestion du service clientèle. La collaboration extrait le contrat du client en tant qu'objet métier d'une application de gestion de fiches clients, puis elle utilise les informations du contrat pour envoyer un objet métier facture à une application de comptabilité.

Une telle collaboration est susceptible d'utiliser trois ports, un pour chacun des connecteurs avec lesquels elle interagit. Chaque port est associé à un type d'objet métier particulier, comme le montre la figure 19 :

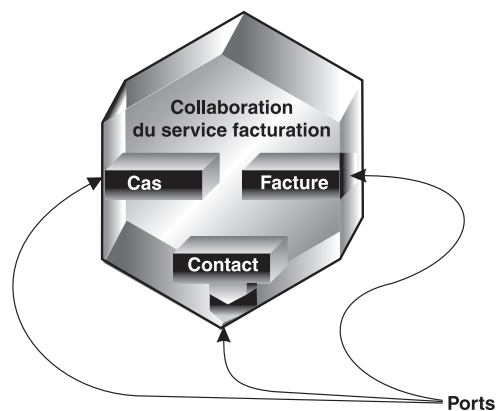


Figure 19. Les ports de la collaboration du service de facturation

Lors de la configuration, l'administrateur crée un objet de collaboration qui contient les ports du modèle. L'administrateur **associe** les ports pour cet objet de collaboration particulier, en associant chaque port à un connecteur ou à un autre

objet de collaboration. Pour certaines collaborations (celles dont le modèle accepte l’instruction Extraire), un port peut au contraire être configuré en tant qu’*Externe*, afin de pouvoir recevoir un objet métier de déclenchement à partir d’une requête d’accès (un appel externe d’une interface d’accès au serveur).

Les ports autorisent la communication entre les entités liées, afin de permettre aux objets de collaboration d’accepter les objets métiers qui déclenchent ses processus métier, et ainsi envoyer et recevoir des objets métier comme requêtes et réponses.

La figure 20 illustre une collaboration fictive, dont la liaison du port de déclenchement accepte l’objet métier Dossier en tant qu’appel provenant de processus externes, et dont les autres ports sont reliés à des connecteurs qui utilisent les objets métier Contrat et facture.

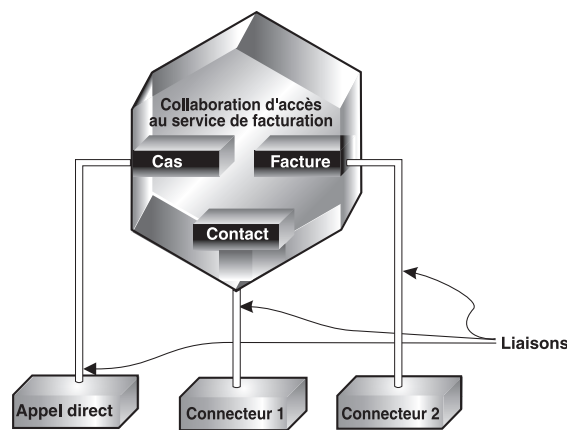


Figure 20. Ports liés aux connecteurs

Un ordinateur dispose de plusieurs interfaces matérielles pour réaliser une analogie. Pour connecter des périphériques à un ordinateur, tels que des haut-parleurs ou un moniteur, vous devez brancher des câbles et des fils dont les formes des connecteurs correspondent aux prises des interfaces de l’ordinateur. La “forme” d’un port de collaboration correspond au type d’objet métier qu’il peut transférer. Un administrateur ne peut associer un port qu’à un connecteur qui prend charge des types d’objets métier spécifiques, de la même façon que vous ne pouvez brancher un câble d’ordinateur que sur l’interface dont la forme correspond.

Une collaboration ne peut être associée à une autre collaboration que si chacune d’entre-elles dispose d’un port qui prend en charge le même type d’objet métier. La figure 21 illustre deux objets de collaboration associés à des ports qui prennent en charge l’objet métier Élément.

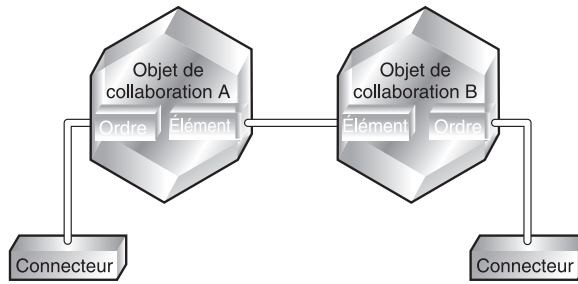


Figure 21. Liaison de deux objets de collaboration

Le nom des ports indique souvent le rôle de l'objet métier pris en charge dans la collaboration. Par exemple, si la collaboration reçoit des informations client et génère des informations contractuelles, ses ports pourraient s'appeler `EntréeClient` et `SortieContrat`.

Intervention sur appel dynamique

Les collaborations implémentent des interventions sur appel avec liaisons explicites, ce qui signifie que l'intervention sur appel de la collaboration doit être configurée pour se connecter à un composant connu, tel qu'un connecteur ou une autre collaboration, alors même que la collaboration est en cours de développement à partir du modèle de collaboration. La liaison explicite est statique, ce qui signifie qu'en cas de modification d'affectation des ports, les objets de collaboration doivent être redéployés dans InterChange Server avant de pouvoir être utilisés. Par ailleurs, les collaborations ne peuvent être déployées tant que tous les ports ne sont pas liés.

InterChange Server comprend une interface de programme d'application qui autorise les interventions sur appel dynamiques, afin de permettre à la collaboration d'effectuer une intervention sur appel vers une destination qui n'avait pas été prédéfinie et liée explicitement pendant le développement de la collaboration. L'API d'intervention sur appel dynamique relève toutes les informations utiles sur le client afin de permettre à une collaboration d'interagir avec de nombreux connecteurs, et le nombre et le type des connecteurs peut ainsi changer dynamiquement. Lors de la création d'une collaboration destinée à recevoir une intervention sur appel dynamique, un port de communication entrante doit être lié en tant que port externe lors de la liaison du port avec l'outil de gestion du système (System Manager), afin de permettre à la collaboration de recevoir l'objet métier de déclenchement par le port externe. L'interface de programme d'application peut transmettre les informations essentielles sur le client, telles que le nom et le type du client. L'interface de programme d'application peut également utiliser les informations relatives à l'adaptateur source pour déterminer la destination de la demande de service. L'interface de programme d'application de l'intervention sur appel dynamique est synchrone.

L'interface de programme d'application de l'intervention sur appel dynamique permet de lier dynamiquement des services sans l'aide d'aucun outil. L'interface de programme d'application et l'ensemble de paramètres associés peuvent être utilisés dans les modèles de collaboration avec un minimum de modifications et de conséquences sur les collaborations existantes. L'interface de programme d'application elle-même ne gère pas les exceptions et ne définit pas de comportement en cas d'anomalie, ces problèmes doivent donc être pris en compte dans le modèle de collaboration. L'interface de programme d'application ne fournit pas non plus de qualité de service, telle que la compensation du support des

transactions. Consultez le *Collaboration Development Guide* pour des informations sur les interfaces de programme d'application.

L'intervention sur appel dynamique prend en charge de façon limitée ou pas du tout les éléments suivants :

- Les interventions sur appel dynamique en provenance d'une collaboration vers une autre collaboration ne forment pas un groupe de collaborations, car aucune liaison explicite de ports n'existe entre les collaborations.
- Les interventions sur appel dynamique ne prennent pas en charge la compensation, il n'y a donc aucune sémantique d'annulation pour les interventions sur appel dynamiques.
- Une collaboration LLBP (processus métier à longue échéance) ne peut effectuer qu'une intervention sur appel dynamique vers un connecteur, et elle doit utiliser le protocole JMS comme type de transfert. Une collaboration LLBP ne peut pas recevoir d'intervention sur appel dynamique, car une collaboration LLBP ne peut pas être liée à un port externe.

Scénarios

À l'intérieur d'une collaboration, un ou plusieurs **scénarios** contiennent le "code" qui implémente un processus métier. Chaque scénario indique ce qui doit se passer en réponse à l'arrivée d'un type particulier d'objets métier qui représentent des types d'événements particuliers. Les scénarios contiennent tout le travail de traitement d'une collaboration.

La relation entre les collaborations et les scénarios est la même que celle qui existe entre les programmes et les routines traditionnels. Les routines permettent au programmeur de scinder la logique d'un programme d'un nombre de manières illimité. Par exemple, un programme peut contenir une grosse routine qui gère plusieurs arguments d'entrée, ou plusieurs routines qui contiennent chacune un argument d'entrée différent.

D'une façon similaire, un développeur de collaborations peut utiliser un scénario ou plusieurs scénarios pour exécuter le travail d'une collaboration. Une collaboration qui synchronise les employés, par exemple, pourrait être conçue de l'une des deux façons suivantes :

- Un scénario gère les notifications d'événements Employé.Créer, Employé.Supprimer, et Employé.MiseÀJour.
- Un scénario gère Employé.Créer, un autre Employé.Supprimer, et un autre Employé.MiseÀJour.

La figure 22 illustre ces deux approches.

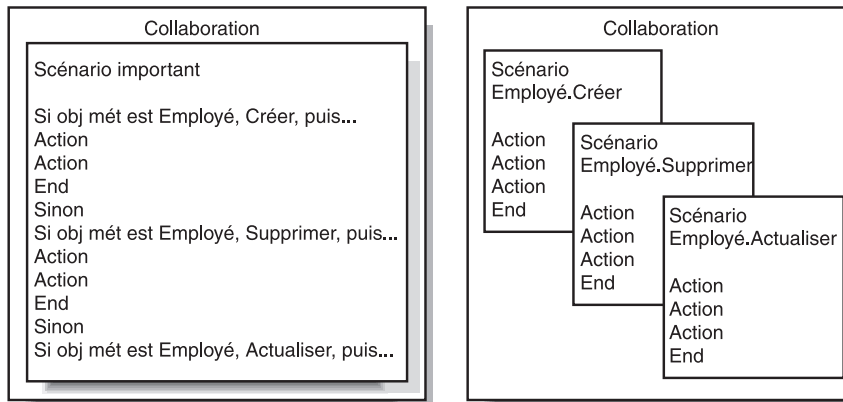


Figure 22. Les scénarios dans les collaborations

Bien que les scénarios s'exécutent indépendamment les uns des autres, vous ne pouvez pas les configurer et les gérer de façon indépendante. Vous configurez la collaboration, et tous les scénarios de la collaboration héritent de ces paramètres.

Logique du processus métier

Les scénarios contiennent la logique des processus métier des collaborations. Un exemple simplifié à l'extrême, figure 23, illustre la logique dans un scénario fictif. La collaboration duplique les modifications des informations relatives aux employés dans une application source en les déplaçant vers une application cible. Ce scénario gère spécifiquement l'instruction Créer.

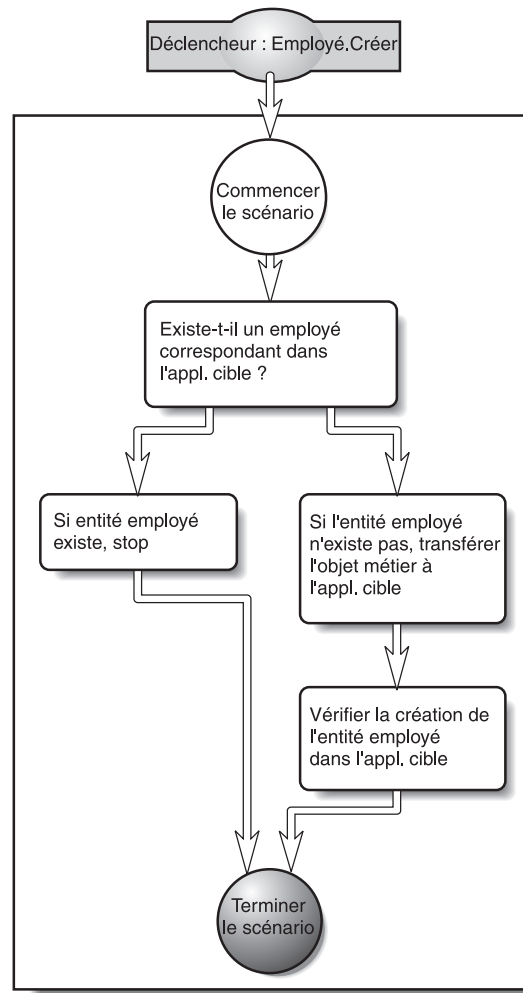


Figure 23. Présentation globale d'un scénario

Dans la figure 23, le déclencheur du scénario est l'objet métier de notification d'événement `Employé.Créer`. Lorsque l'objet métier `Employé.Créer` arrive, le scénario commence. Tout d'abord, le scénario vérifie si les informations sur l'employé existent déjà dans l'application cible.

Si les informations sur l'employé existent déjà dans l'application cible, le scénario s'arrête. Si les informations sur l'employé n'existent pas encore, le scénario envoie l'objet métier de requête `Employé.Créer` au connecteur pour l'application cible. Une étape de vérification permet de s'assurer que l'opération s'est déroulée avec succès.

La logique d'un processus métier dans un scénario réel implique un nombre d'étapes bien plus important, dont chacune implémente une action de la collaboration. Voici quelques exemples des types d'actions qu'une collaboration pourrait contenir :

- Obtenir le type, les valeurs d'attribut, ou l'instruction d'un objet métier reçu
- Créer un nouvel objet métier, soit sans valeurs, soit en clonant un objet métier existant
- Comparer une valeur d'attribut avec une constante ou avec une autre valeur d'attribut
- Comparer deux objets métier pour vérifier s'ils sont identiques

- Envoyer un objet métier à un connecteur ou à une autre collaboration pour demander une opération ; traiter le résultat de l'opération
- Obtenir la valeur de l'une des propriétés configurables de la collaboration
- Créer un journal d'informations, d'avertissements ou de messages d'erreur

Interactions avec les connecteurs et les applications

Grâce à ses ports, une collaboration interagit avec ses connecteurs, avec l'interface d'accès et avec d'autres collaborations. La collaboration reçoit son déclenchement, envoie des requêtes et obtient des réponses à travers ces ports. C'est à ce moment que s'effectue le transcodage en opérations d'applications.

En utilisant le scénario représenté dans la figure 23 en tant qu'exemple simplifié, la figure 24 décrit la façon dont la collaboration conduit des opérations à travers ses ports lors d'une interaction publication-abonnement. La figure 24 ne montre qu'un exemple individuel de chemin d'exécution, le chemin d'accès qui est pris lorsque l'application cible ne contient pas encore les informations concernant l'employé. Les chemins d'accès gris fléchés en figure 24 montrent les transcodages réalisés par le connecteur B lorsqu'il transfère des requêtes de collaboration en requêtes vers l'interface de programme d'application et qu'il transfère les réponses de l'application vers ses propres réponses à la collaboration.

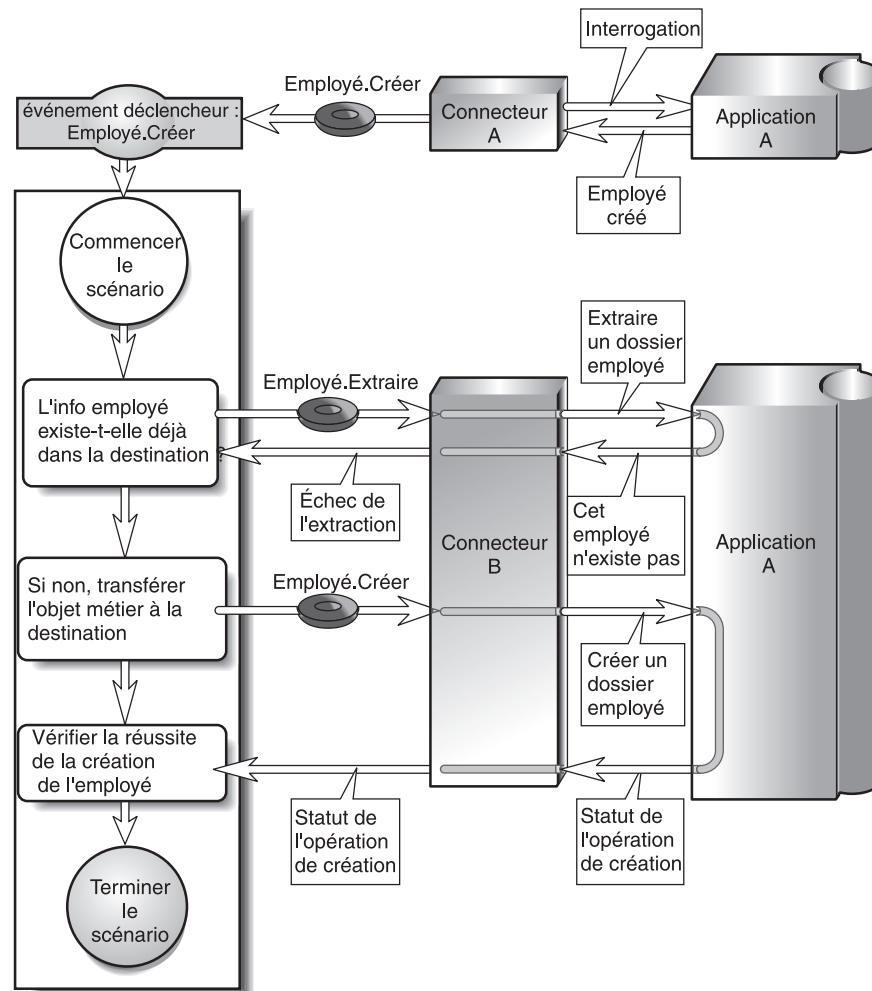


Figure 24. Traitement en contexte d'exécution

Dans cette figure, la représentation du connecteur comprend à la fois les fonctions de l'agent connecteur et celles de l'agent contrôleur.

La figure 24 illustre le processus suivant :

1. Lorsque l'application A crée un nouveau dossier d'employé, le connecteur A l'extrait.
2. Le connecteur A crée un objet métier `Employé.Créer` et envoie cette notification d'événement à la collaboration où son arrivée déclenche le démarrage de ce scénario.
3. Ce scénario vérifie si les informations relatives à l'employé sont réellement nouvelles en essayant d'extraire les informations de ce même employé de l'application cible. Pour ce faire, le scénario envoie une requête `Employé.Extraire` au connecteur B.
4. Le connecteur B tente d'extraire les informations relatives à l'employé mais apprend que l'employé n'existe pas dans l'Application B. Le connecteur envoie alors un statut d'échec à la collaboration.
5. Le scénario passe maintenant à l'envoi de la requête `Employé.Créer`. Lorsque le connecteur B reçoit la requête `Employé.Créer`, il utilise l'interface de programme d'application pour créer le nouvel employé.

6. Le connecteur B renvoie un statut de succès pour indiquer que l'opération a réussi.
7. Le scénario reçoit le statut de succès et s'arrête.

Initialisation de la collaboration

Lors de la conception, le développeur de la collaboration spécifie les objets métier dont les réceptions par la collaboration vont déclencher l'exécution de chaque scénario. Lorsque vous configurez et activez une collaboration, la collaboration s'initie. La collaboration s'abonne auprès de tous les objets métier qui déclenchent ses scénarios, puis elle se met en attente.

Lorsqu'un agent connecteur envoie à son contrôleur l'un des objets métier auquel il est abonné :

1. Le contrôleur du connecteur transfère l'objet métier à la collaboration.
2. La collaboration appelle le scénario approprié, qui commence alors l'exécution.

Une fois que vous avez configuré et activé une collaboration, vous n'avez pas besoin de l'activer de nouveau tant que vous ne l'avez pas désactivée explicitement. Si InterChange Server s'arrête et redémarre, il redémarre également toutes les collaborations qui étaient précédemment actives.

Récapitulatif

Ce chapitre vous a présenté certaines fonctions des collaborations. Les points principaux à retenir sont :

- Vous créez des objets de collaboration à partir de modèle de collaboration et vous les liez à des connecteurs ainsi qu'à d'autres objets de collaboration pour les rendre exécutables.
- Les collaborations représentent indirectement des objets métier, par l'intermédiaire des ports. Vous configurez un objet de collaboration en liant chacun de ses ports à une définition d'objet métier et à un connecteur ou à un autre objet de collaboration ou à des appels externes.
- Les scénarios implémentent la logique de processus métier des collaborations. Les scénarios sont des gestionnaires d'événements qui s'exécutent à partir de l'arrivée d'objets métier. Lorsque les scénarios s'exécutent, ils envoient des objets métier aux connecteurs tout en recevant des objets métier de ces connecteurs.
- Pour activer une collaboration à exécuter, il vous suffit de la configurer et de la démarrer. La collaboration s'abonne aux types d'objets métier qui peuvent déclencher ses scénarios. Lorsque l'un de ces objets métier arrive, la collaboration exécute le scénario approprié.

Le chapitre suivant étudie plus en détail les objets métier.

Chapitre 4. Objets métier

Les objets métier transportent la signification des processus métier d'une application à une autre, ils traversent les collaborations et les connecteurs en effectuant à la fois des requêtes de données et d'actions. Ce chapitre s'intéresse à la structure et aux composants des objets métier. Les sections suivantes seront abordées :

- «Définitions des objets métier et les objets métier»
- «Composants d'une définition d'objet métier» à la page 60
- «Présentation détaillée des objets métier propres à une application» à la page 62
- «Les options de modification» à la page 64
- «Récapitulatif» à la page 65

Définitions des objets métier et les objets métier

Le Chapitre 1, «Présentation d'IBM WebSphere InterChange Server», à la page 1, introduit les objets métier sans distinction entre les définitions des objets métier et les instances des objets métier eux-mêmes. Il est temps d'étudier ces différences :

- Une **définition d'un objet métier** indique le type et l'ordre des informations dans chaque entité gérée par IBM WebSphere InterChange Server, et les instructions qui sont prises en charge. Le référentiel InterChange Server stocke les définitions des objets métier.
- Un **objet métier** est une instance de la définition, contenant les données en cours de traitement. Les objets métier sont créés en phase d'exécution et ne sont pas stockés dans le référentiel.

La figure 25 illustre la relation entre la définition de l'objet métier et l'objet métier lui-même.

Par exemple, une définition d'objet métier peut contenir des attributs pour le nom de l'employé, son adresse, son ID employé et toute autre information pertinente. Les attributs d'un objet métier sont similaires aux champs d'un formulaire ou aux colonnes d'une table de base de données.

Un attribut peut également faire référence à un objet métier enfant ou à un ensemble d'objets métier enfants, tel qu'un ensemble de lignes articles dans un contrat ou des références de pièces détachées dans une facture.

Attribut ObjectEventId

L'attribut ObjectEventId est un attribut obligatoire qui se trouve en dernière position dans chaque objet métier. Lorsqu'un connecteur publie un événement, il utilise l'attribut ObjectEventId de la définition de l'objet métier pour stocker une valeur unique qui identifie l'instance spécifique de l'objet métier qui est créé.

La valeur de l'attribut ObjectEventId est générée et gérée par le système, qui l'utilise pour identifier et suivre le flux de l'événement spécifique à travers le système. Le développeur ne doit pas mapper l'attribut ObjectEventId ni l'alimenter par l'intermédiaire d'un agent du connecteur ou d'un gestionnaire de données.

Types d'attributs simples et composés

Si le type de l'attribut est un type de données de base, tel que Chaîne, Booléen, Double, Flottant ou Entier, la valeur de cet attribut est une donnée discrète, telle la valeur d'un champ dans une base de données. Un tel attribut est souvent qualifié de **simple**. Par exemple, LastName, CustomerID, PartNumber, AssignedTo, et Price sont des attributs simples.

Si le type d'un attribut est le nom d'une autre définition d'objet métier (de type composé), la valeur de l'attribut est un objet métier enfant ou un ensemble d'objets métier enfants. Un tel attribut est qualifié de **composé**. Customer, Contract, et Oracle_Contact sont des exemples d'attributs composés.

Propriétés de l'attribut

Un ensemble de propriétés **propriétés** définissent la valeur que l'attribut représente. Sans toutefois présenter toutes les possibilités de propriétés, la figure 26 illustre la place des propriétés des attributs dans la définition d'un objet métier.

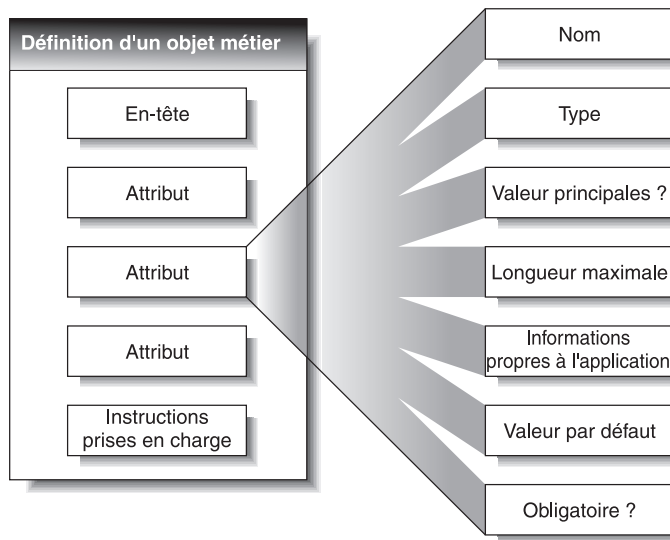


Figure 26. Propriétés des attributs

L'ensemble des propriétés pour un attribut particulier dépend du type de l'attribut (simple ou composé). En effet, les propriétés d'un attribut sont différentes selon qu'il se réfère à une unité simple de données ou à un objet métier enfant.

Instructions

Les **instructions** indiquent les actions à effectuer sur les données de l'objet métier. La définition d'un objet métier contient une liste d'instructions. Un objet métier ne contient qu'une seule instruction. Les verbes les plus courants dans les définitions d'objets métier sont : Create, Retrieve, Update, et Delete (créer, extraire, mettre à jour et supprimer). La signification d'une instruction dépend du rôle de l'objet métier. L'instruction peut décrire un événement d'application, faire un appel, effectuer une requête, ou identifier le résultat d'une requête précédente.

Remarque : Certaines applications ne prennent pas en charge les requêtes de suppression physique. Pour ce type d'applications, le système IBM WebSphere InterChange Server réalise la suppression logique équivalente, qui est généralement une mise à jour de l'état à inactif. De plus, même si une application prend en charge les suppressions physiques, vous pouvez configurer le système IBM WebSphere InterChange Server pour qu'il convertisse l'instruction Delete (supprimer) en instruction Update (mettre à jour) lorsque des requêtes sont envoyées à cette application.

Présentation détaillée des objets métier propres à une application

Un objet métier propre à l'application contient les données qu'un agent du connecteur transfère vers et à partir d'une application particulière. Par conséquent, chaque définition d'objet métier propre à l'application reflète le modèle de données de l'application et la méthode d'accès de l'agent du connecteur.

Même lorsque deux objets métier propres à une application se réfèrent à des entités d'applications similaires, il existe des différences dans la façon dont les attributs sont organisés ainsi que dans les informations propres à l'application qu'ils contiennent.

Organisation des attributs

Les applications organisent souvent les mêmes informations de façons différentes. Par exemple, l'Application A stocke le numéro de téléphone et de télécopieur d'un contact dans quatre champs, tandis que l'Application B stocke les mêmes numéros dans deux champs.

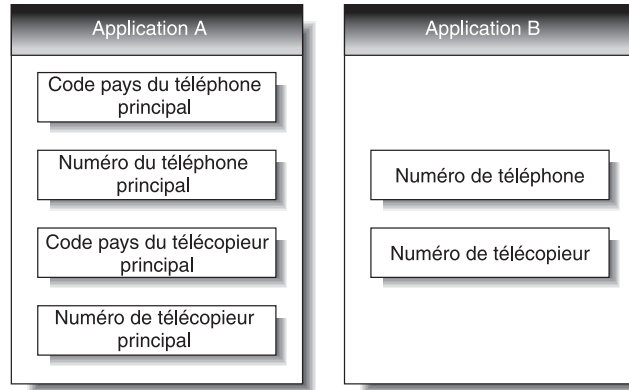


Figure 27. Données téléphoniques dans deux applications

Les définitions d'objet métier pour l'objet métier de l'Application A et pour l'objet métier de l'Application B ont des attributs différents pour refléter cette différence.

Informations propres à l'application

Les objets métier propres à une application diffèrent également par le fait qu'ils peuvent contenir de façon facultative des instructions de traitement intégrées pour leur agent du connecteur, appelées informations propres à l'application. Les informations propres à l'application peuvent être constituées de toute information dont l'agent du connecteur a besoin pour traiter l'objet métier.

La définition d'un objet métier peut contenir des informations propres à l'application qui s'appliquent à l'intégralité de l'objet métier, à chaque attribut et à chaque instruction. Chaque information propre à l'application qui apparaît dans la définition de l'objet métier fournit des informations que le connecteur utilise lors de ses interactions avec l'application.

Informations propres à l'application pour un objet métier

Les informations propres à l'application pour l'objet métier fournissent des informations que l'agent du connecteur utilise lorsqu'il procède au traitement de l'objet métier dans son ensemble.

Informations propres à l'application pour un attribut

Généralement, les informations propres à l'application qui s'appliquent à un attribut identifient l'emplacement de la valeur de l'attribut dans l'application. L'agent du connecteur utilise cet identificateur lorsqu'il construit des appels API à destination des applications pour extraire ou entrer la valeur d'attribut.

Les informations propres à l'application prennent des formes différentes pour les différentes applications. Il arrive parfois que l'agent du connecteur puisse trouver l'emplacement de l'attribut en se référant au format et aux noms des champs de l'application. D'autres fois, la référence à l'emplacement est plus complexe à trouver.

Le tableau 4 illustre deux exemples d'informations propres à l'application pour un objet métier Client.

Tableau 4. Informations propres à l'application pour des modèles d'attributs

Attribut	Informations propres à l'application pour un modèle d'application 1	Informations propres à l'application pour un modèle d'application 2
ID client	CUST:CID	cust:site_id::3::
Nom du client	CUST:CNAM	cust:name::3::
Statut	CUST:CSTAT	cust:status::1:0:Status
Secteur d'activité	CUST:CIND	cust:industry_type::3::Industry

Ces valeurs ont les significations suivantes :

- La valeur du modèle d'application 1 indique un champ dans la table CUST.
- La valeur du modèle d'application 2 indique une table, un champ et une relation dans la base de données de l'application, selon le format suivant :

table:champ:relation:type:valeur
par défaut

où *type* est égal à 1 (int), 2 (float), ou 3 (string)

Dans certains cas exceptionnels, des informations propres à l'application sont nécessaires pour les attributs. Par exemple, certaines applications fournissent des désignations très simples et faciles à utiliser pour les unités de données. Imaginez une application qui identifie les modèles de champs comme cela est illustré dans le tableau 5.

Tableau 5. Identificateurs de modèles d'applications

Attribut	Identificateur de l'application pour le champ contenant la valeur
ID client	XCustomerID
Nom du client	XCustomerName
Statut	XStatus
Secteur d'activité	XIndustry

Dans l'exemple illustré dans le tableau 5, il est aisé pour l'agent du connecteur d'associer un attribut avec son identificateur dans l'application, car les règles de conversion sont classiques : ajouter le X ou enlever le X. Par conséquent, les attributs pour les objets métier de cette application ne nécessiteront probablement pas d'informations propres à l'application.

Informations propres à l'application pour les instructions

La définition d'un objet métier peut comporter des informations propres à l'application pour chaque instruction qu'elle prend en charge. L'information propre à l'application indique à l'agent du connecteur la façon de traiter l'objet métier lorsque cette instruction est activée.

Les options de modification

Vous pouvez modifier soit la définition des objets métier propres à l'application, soit la définition des objets métier génériques, et il peut être nécessaire de modifier ces deux types de définitions. Par exemple, pour modifier une collaboration pour qu'elle traite des données applicatives supplémentaires, vous devez modifier les deux types de définitions d'objets métier, pour que chaque objet métier puisse

prendre en charge ces données. Lorsque vous modifiez la définition d'un objet métier, vous devez également modifier les composants de mappage.

Récapitulatif

Ce chapitre vous a présenté la définition et l'utilisation des objets métier dans le système IBM WebSphere InterChange Server. Les principaux points à retenir sont :

- Les définitions d'objets métier sont les spécifications qui définissent le type et l'ordre des données applicatives et l'ensemble des instructions qui peuvent déclencher des opérations sur les données. Un objet métier est une instance d'une définition qui contient les données en cours de traitement et une instruction.
- Les objets métier propres à une application pour les mêmes types d'entités sont différents, car les modèles d'applications sont différents. Les principales différences concernent l'ordre et le nombre d'attributs, ainsi que les informations propres à l'application.
- Les définitions d'objets métier contiennent des instructions et des attributs. Les attributs dont le type des données est simple se réfèrent aux données, tandis que les attributs dont le type des données est composé se réfèrent à des objets métier enfants.
- À l'intérieur de la définition de l'objet métier, les informations propres à l'application contiennent des données applicatives pertinentes. L'agent du connecteur utilise les informations propres à l'application pour conduire ses interactions avec l'application.
- Vous pouvez modifier à la fois les objets métier génériques et les objets métier propres à une application pour modifier le flux des données sur un site.

Chapitre 5. Connecteurs

Un **connecteur** sert d'intermédiaire entre une application (ou toute autre entité par programmation, tel un serveur Web) et une ou plusieurs collaborations. Un connecteur peut être propre à une application (telle que SAP R/3 version 4) ou à une technologie, telle qu'un format de données ou un protocole (XML ou EDI). La communication entre un connecteur et des collaborations peut prendre les deux formes suivantes :

- **Notifications d'événements** d'applications, que les connecteurs propres à des applications transmettent aux collaborations, et
- **Traitement des requêtes**, que les connecteurs réalisent au nom des collaborations.

Les connecteurs peuvent être configurés pour des applications qui résident sur le réseau local, et pour des applications éloignées qui résident au-delà d'un pare-feu Internet.

La plupart des connecteurs partagent certains comportements, qui ne diffèrent que par leur manière d'interagir avec les applications et avec les objets métier propres aux applications. Ce chapitre va vous présenter à la fois les comportements communs des connecteurs et leurs différences. Les sections suivantes seront abordées :

- «Initialisation du connecteur»
- «Notification d'événement» à la page 69
- «Traitement des requêtes» à la page 75
- «Capacités de traitement simultané» à la page 78
- «Construction et déconstruction des objets métier» à la page 78
- «Configuration du connecteur» à la page 81
- «Développement du connecteur» à la page 82
- «Récapitulatif» à la page 83

Les connecteurs pour de nombreuses applications et technologies font partie de la livraison du produit IBM WebSphere Business Integration Adapters. Un adaptateur WebSphere Business Integration comprend un connecteur, ses fichiers de messages et son outil de configuration, et il peut également inclure les mécanismes de création et de gestion des objets métier propres à ce connecteur. Cependant, si vous devez créer un connecteur personnalisé ou modifier un connecteur existant, il vous faudra connaître de façon détaillée le comportement des connecteurs. Pour plus d'information sur la façon de créer un connecteur personnalisé, consultez les manuels suivants : *Connector Development Guide for Java* ou *Connector Development Guide for C++*.

Initialisation du connecteur

Dans un système InterChange Server, un connecteur est constitué des composants suivants, qui doivent tous être démarrés pour permettre au connecteur de s'exécuter :

- Un **contrôleur de connecteur** démarre avec le serveur InterChange Server ; si le connecteur a été configuré, InterChange Server lui crée automatiquement un contrôleur.

- Un **agent du connecteur** doit être démarré explicitement, soit par une ligne de commande soit par un raccourci. Cependant, vous pouvez configurer une option de l'agent du connecteur pour qu'il redémarre automatiquement après un arrêt anormal ou une panne.

Un agent du connecteur communique avec un serveur InterChange Server particulier lors du démarrage et pendant toute la durée de l'exécution de cet agent du connecteur. Lorsque vous démarrez un agent du connecteur, vous lui indiquez le nom d'un serveur InterChange Server. La première opération de l'agent du connecteur est de contacter le serveur InterChange Server pour lui demander ses propres paramètres de configuration et la définition des objets métier qu'il est configuré pour prendre en charge.

Un serveur InterChange Server ne doit pas obligatoirement être en cours de fonctionnement lorsque vous démarrez l'agent du connecteur. Cependant, l'agent du connecteur ne peut effectuer aucune tâche utile tant qu'il n'obtient pas les informations nécessaires à son initialisation. Si vous démarrez un agent du connecteur alors que le serveur InterChange Server n'est pas en cours de fonctionnement, l'agent du connecteur tente de façon répétée de contacter le serveur.

Avant qu'un agent du connecteur puisse se connecter à une application, il doit recevoir sa configuration et télécharger ses définitions d'objets métier du référentiel InterChange Server. Il obtient également une liste des souscriptions d'événements de collaborations qu'il doit prendre en charge.

La figure suivante illustre les trois étapes du démarrage de l'agent du connecteur.

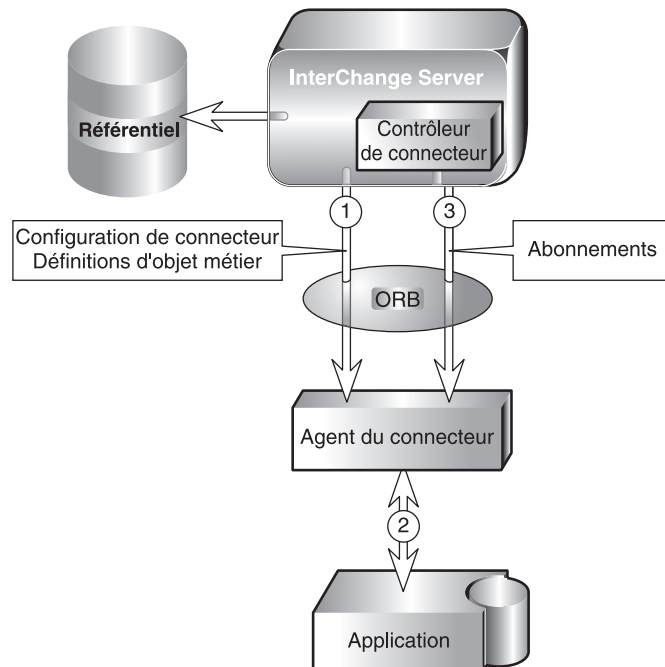


Figure 28. L'initialisation du connecteur

L'agent du connecteur effectue les tâches relatives à son initialisation dans l'ordre suivant :

1. Il contacte InterChange Server pour les informations nécessaires à sa configuration et la définition des objets métier.
2. Il se connecte à l'application.
3. Il contacte le contrôleur de connecteur pour obtenir la liste des abonnements.

Un connecteur télécharge la définition de ses objets métier et certaines de ses propriétés de configuration, telles que son nom d'utilisateur de l'application et son mot de passe, uniquement lors de l'initialisation. La valeur de certaines autres propriétés, telle la propriété qui contrôle la fonction de trace du connecteur, peut être changée dynamiquement.

Notification d'événement

De nombreux connecteurs (pas tous) engagent une interaction publication-abonnement avec les collaborations et utilisent la notification d'événement pour déclencher des événements dans ces collaborations. Un connecteur dont l'application fournit des événements déclencheurs aux collaborations InterChange Server doit prendre connaissance de ces événements et envoyer les données correspondantes à la collaboration abonnée. La figure suivante illustre les interactions d'un connecteur relatives à la notification d'événement.

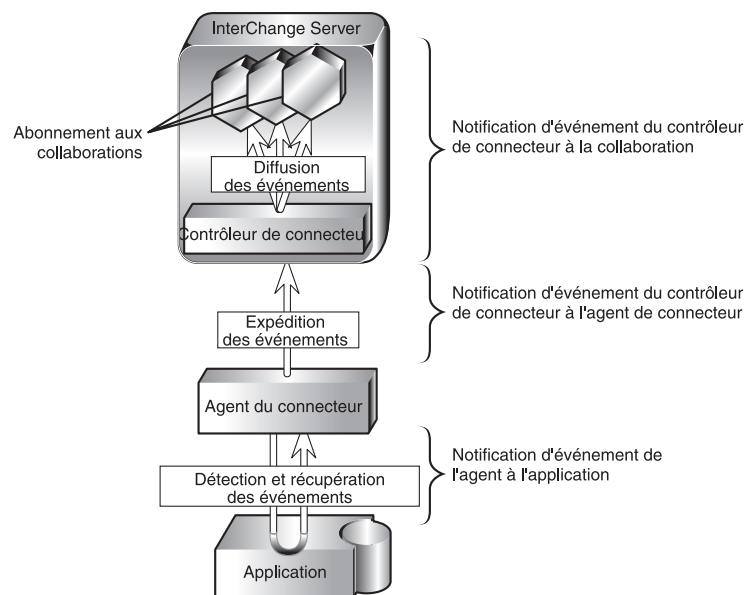


Figure 29. Notification d'événement dans un connecteur

La façon dont l'agent du connecteur détecte et extrait les événements diffère d'un connecteur à un autre. Cependant, la façon dont l'agent du connecteur envoie les événements aux contrôleurs de connecteurs, et la façon dont les contrôleurs de connecteurs délivrent ces événements aux collaborations est identique pour tous les connecteurs.

Les sous-sections suivantes décrivent les concepts généraux relatifs au fonctionnement de la plupart des connecteurs, à savoir :

- La façon dont les connecteurs utilisent les mécanismes de notification d'événements des application, et
- La façon dont les connecteurs détectent et traitent les événements.

Il ne s'agit pas ici de décrire l'implémentation spécifique d'un connecteur particulier.

Configuration du mécanisme de notification d'événements de l'application

Pour un connecteur, un **événement d'application** est une opération quelconque qui modifie les données d'une entité d'application qui est associée à la définition de l'objet métier. Il existe d'autres types d'événements dans les applications. Par exemple, le clic d'une souris est un événement pour le système de fenêtrage ou l'interface de formulaires. Cependant, le connecteur ne s'intéresse qu'aux événements qui concernent les données, comme la création, la mise à jour, la suppression ou tout autre opération qui modifie le contenu du magasin de données de l'application.

Certaines applications interceptent et rapportent les événements de façon explicite, offrant ainsi une gestion conviviale des événements et la possibilité de configurer le texte associé à l'événement. D'autres applications, dépourvues du concept d'événement ponctuel et rapportable, peuvent silencieusement mettre à jour leurs bases de données lorsqu'il se passe quelque chose.

Pour la plupart des connecteurs, la personne chargée de la configuration de l'application doit mettre en place un **mécanisme de notification d'événement** à l'usage des connecteurs. Un mécanisme de notification d'événement est une liste séquentielle d'opérations qui s'exécute dans l'application. Elle peut prendre l'apparence physique de la file d'attente des événements d'une application, de la boîte de réception d'une messagerie électronique, ou d'une table d'une base de données.

Quels types de mécanismes de notification d'événement les connecteurs utilisent-ils ? Les sections suivantes illustrent quelques approches générales.

Lorsque les applications prennent en charge les événements

Si une application est basée sur des événements, elle dispose probablement d'une interface de notification d'événements à l'usage des applications client tels les connecteurs. L'application peut également permettre aux personnes chargées de la mise en oeuvre de configurer le texte du rapport d'événement. Pour de telles applications, la configuration du mécanisme de notification d'événements du connecteur fait partie intégrante de la configuration de l'application.

Par exemple, imaginez qu'une application vous permette d'installer un script qui s'exécute lorsqu'un type particulier d'événement se produit et que ce script peut déposer une notification dans une boîte de réception d'événements. Pour installer le connecteur de cette application, vous devez créer un compte utilisateur pour le connecteur, écrire ou obtenir des scripts pour gérer les événements que vous souhaitez suivre, installer les scripts, spécifier le type d'événement qui déclenche chaque script, et créer la boîte de réception. Une fois ces opérations terminées, l'agent du connecteur extrait périodiquement le contenu de la boîte de réception pour vérifier s'il y a de nouveaux événements.

La figure suivante illustre une configuration d'applications qui intègre une boîte de réception.

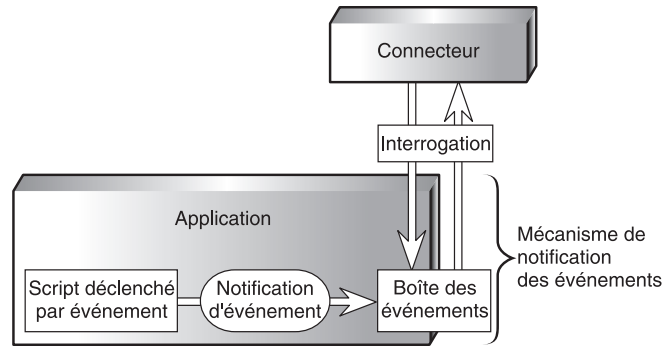


Figure 30. Exemple d'utilisation d'une boîte de réception d'événements pour la notification d'événements

Une autre application pourrait disposer d'un système de flux de travaux interne qui générerait des courriels ou écrirait à une file d'attente des événements lorsqu'une opération particulière se produit. La figure 31 illustre une application qui dispose d'un référentiel d'objet métier dans lequel les objets métier et les événements sont définis. Dans cette figure, Customer (client) est l'objet métier d'une application, Delete (supprimer) et Update (mettre à jour) sont les types d'événements qui lui sont associés.

Lorsqu'un événement d'objet métier tel que Customer.Update se produit, l'événement est envoyé au système des flux de travaux, qui dépose une entrée dans une table des événements de la base de données de l'application.

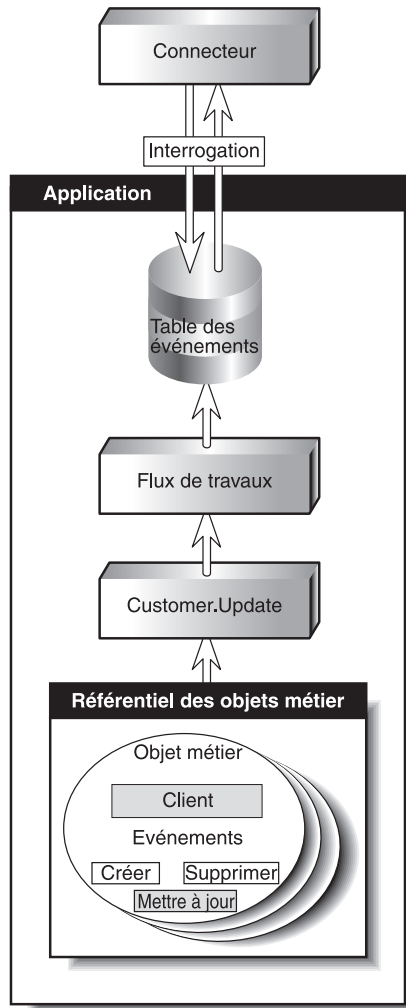


Figure 31. Exemple d'utilisation des flux de travaux d'une application pour la notification d'événements

Lorsque les applications ne prennent pas en charge les événements

L'interface de programme d'application est la méthode d'interaction privilégiée des connecteurs pour interagir avec les événements d'applications, car cette interface fournit une structure permettant d'imposer le modèle de données et la logique de l'application. Cependant, certaines interfaces de programmation d'applications ne fournissent pas en natif la prise en charge de la notification d'événements.

L'une des façons permettant à un connecteur de recevoir des notifications d'événements provenant de telles applications est d'interagir avec la base de données de l'application. Par exemple, la personne chargée de la configuration peut définir un déclencheur sur une table *Employés* qui détecte les mises à jour des lignes. Lorsqu'une mise à jour se produit, le déclencheur insère une information à propos de la mise à jour dans une table spéciale. Chaque nouvelle ligne qui apparaît dans cette table d'événements représente une notification d'événement. Le connecteur peut utiliser une requête SQL pour extraire les nouveaux événements de la table.

La figure 32 illustre cette approche.

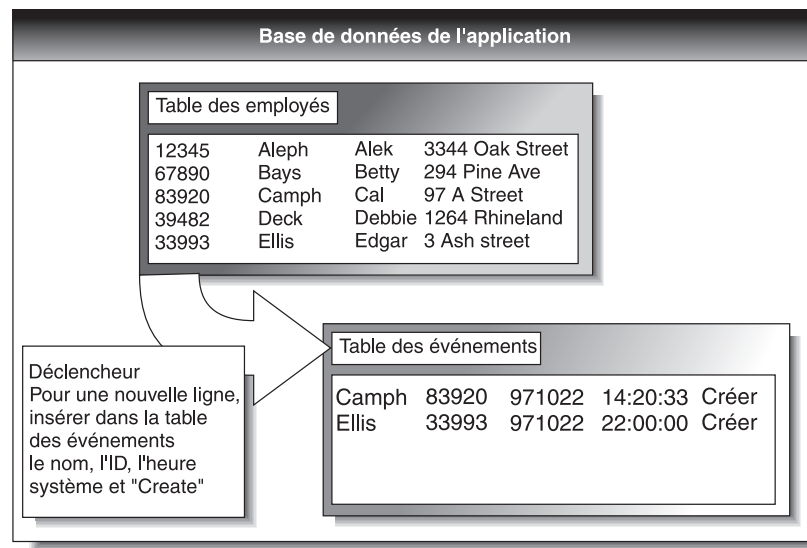


Figure 32. Exemple d'utilisation de la base de données pour la notification d'événements

Dans la figure 32, la base de données de l'application dispose d'un déclencheur lié à la création d'enregistrements dans la table Employés. À chaque fois que l'application insère un nouvel enregistrement, le déclencheur crée une ligne dans la table d'événements. La ligne contient les valeurs principales de l'enregistrement du nouvel employé (nom et identificateur employé), l'heure système et le type d'événement, Créer.

Détection d'un événement

Un agent du connecteur apprend les événements d'applications en interrogeant le mécanisme de notification d'événements pour savoir si de nouveaux événements se sont produits. La méthode de l'interrogation est complètement propre à l'application, elle est basée sur le mécanisme de notification d'événements utilisé par le connecteur.

L'interrogation est configurable. Lorsque vous utilisez System Manager pour configurer un connecteur, vous pouvez :

- Déterminer la fréquence avec laquelle l'agent du connecteur effectue l'interrogation
- Définir les heures pendant lesquelles l'agent du connecteur effectue l'interrogation

Un agent du connecteur n'a pas besoin d'interroger une application si les collaborations ne s'intéressent pas aux événements de l'application. Si une application particulière participe à des collaborations mais ne représente pas la source des événements, vous pouvez désactiver l'interrogation de l'agent du connecteur en définissant sa fréquence d'interrogation à "no."

Traitement d'un événement

Après avoir détecté un événement, l'agent du connecteur :

- Associe l'événement de l'application avec une définition d'objet métier et une instruction qui représente l'événement
- Vérifie les abonnements à l'événement

- Extrait les données applicatives
- Envoie un objet métier au contrôleur de connecteur, si nécessaire
- Archive l'événement (facultatif)

Association d'un événement d'application avec une définition d'objet métier

Lorsqu'un agent du connecteur extrait un événement, il doit déterminer quelles collaborations se sont abonnées à l'événement. Cependant, étant donné que les collaborations s'abonnent à des événements sous la forme *type-objet-métier.instruction*, le connecteur doit tout d'abord déterminer la définition de l'objet métier et l'instruction qui représentent l'événement.

L'agent du connecteur utilise le texte de l'événement pour associer l'événement à une définition d'objet métier et une instruction, comme le montre le tableau 6.

Tableau 6. Structure des textes d'événements et des objets métier

Type de données contenues dans l'événement d'application	Exemples	Utilisation
Type d'entité d'application	Client, Partie, Élément	Déterminer la définition d'objet métier associée
Opération qui s'est produite	Créer, Mettre à jour, Supprimer	Déterminer l'instruction active de l'objet métier

Par exemple, un connecteur peut associer le texte d'événement suivant à l'objet métier Employé.Créer :

```
1997.10.19.12:50.22 employé créé lname="como" id="101961"
```

Le texte de l'événement contient :

- Un horodatage qui permet de donner un identifiant unique à l'événement
- L'entité d'application "employé"
- L'opération "créé"
- Le nom et l'ID de l'employé sont les valeurs principales qui permettent d'extraire le reste des informations relatives à l'employé.

Cet exemple est simplifié, d'autres types de textes d'événement peuvent nécessiter des traitements plus complexes de la part de l'agent du connecteur.

Vérification des abonnements

Après avoir associé l'événement à une définition d'objet métier et à une instruction, l'agent du connecteur vérifie sa liste d'abonnement interne pour savoir si des collaborations sont intéressées par cet événement. La liste des abonnements est toujours à jours car le contrôleur de connecteur met à jour l'agent du connecteur à chaque fois qu'il y a une modification d'abonnement.

Si l'événement ne correspond à aucun abonnement en cours, l'agent du connecteur crée un message d'avertissement dans le journal et élimine l'événement.

Construction d'un objet métier

Si l'événement fait l'objet d'un abonnement, l'agent du connecteur construit un objet métier, utilise les valeurs principales pour extraire les données applicatives, puis complète l'objet métier. La section «Construction et déconstruction des objets métier» à la page 78 décrit et illustre le processus de construction d'un objet métier.

Envoi de l'objet métier au contrôleur de connecteur

Un agent du connecteur envoie un objet métier au contrôleur de connecteur via le mode de transfert utilisé (messagerie ou CORBA). L'agent du connecteur sait uniquement qu'un abonnement est en cours pour l'objet métier, mais il ne sait pas quelles collaborations sont abonnées : le contrôleur de connecteur se charge de la distribution aux collaborations.

Archivage des événements

L'archivage des événements d'applications est utile pour le dépannage et la tenue d'archives. Une archive d'événement contient des informations d'état sur chaque événement, telles que :

- Envoi réussi à InterChange Server
- Aucun abonnement pour cet événement
- Échec du traitement

Si une application fournit une fonction d'archivage des événements, elle est généralement utilisée par le connecteur. Un connecteur pour une application qui ne prend pas en charge l'archivage des événements peut disposer de sa propre fonction d'archivage d'événements. Par exemple, si un mécanisme de notification d'événements est du même type que le mécanisme de base de données illustré en figure 32, un déclencheur de base de données pourrait copier les événements supprimés dans une table d'archives.

Traitement des requêtes

Les connecteurs peuvent recevoir des requêtes des collaborations et effectuer des requêtes auprès de l'application de leur part. Par exemple, une collaboration peut envoyer à un connecteur une requête pour supprimer un contrat, mettre une pièce à jour ou créer un client, sous la forme d'objets métier Contrat.Supprimer, Pièce.MettreAJour, ou Client.Créer.

Lorsqu'un contrôleur de connecteur reçoit une requête d'une collaboration, il transmet la requête à l'agent du connecteur. L'agent du connecteur transcode l'objet métier en requête de l'application (en général, un ensemble d'appels à l'interface de programme d'application), puis renvoie les résultats.

La figure 33 illustre les interactions d'un connecteur dans le cadre de la gestion des requêtes formulées par une collaboration.

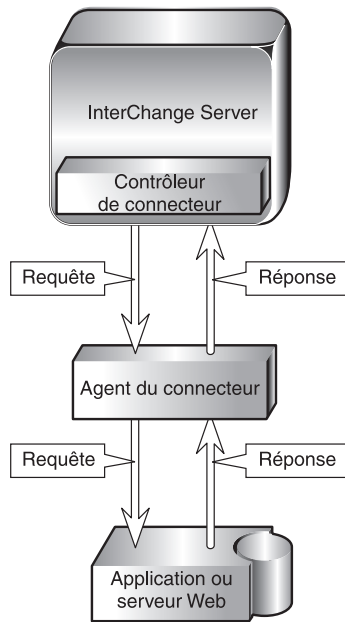


Figure 33. Interactions du connecteur dans la gestion des requêtes

Remarque : Un collaborateur envoie un objet métier à un connecteur spécifique dans une interaction de type requête/réponse, contrairement aux interactions publication/abonnement qui caractérisent la notification d'événement.

Lorsqu'un agent du connecteur reçoit une requête, il va déterminer la façon de traiter la requête à partir de trois types d'informations :

- L'instruction de l'objet métier
- Les informations propres à l'application pour cette instruction
- Les métadonnées contenues dans la définition même de l'objet métier qui sont utilisées lors de la construction et de la destruction de l'objet métier

Ces facteurs sont décrits dans les rubriques ci-dessous.

Traitement à base d'instructions

Un agent du connecteur réagit aux instructions Create, Retrieve, Update, ou Delete d'une requête selon la logique et l'interface de programme d'application de son application. Deux agents du connecteur peuvent gérer différemment le même type de requête, bien que les résultats doivent logiquement être les mêmes.

Pour certains connecteurs, une seule méthode est requise pour réaliser des opérations sur un objet métier, sans tenir compte de l'instruction contenue dans la requête. Mais pour de nombreux connecteurs, chaque instruction requiert une méthode différente.

Lorsqu'un agent du connecteur reçoit une requête, il appelle la méthode de l'application qui correspond à l'instruction active de l'objet métier. Par exemple, lorsqu'un agent du connecteur reçoit un objet métier `AppAEmployee.Update`, il appelle la méthode `Update` sur l'objet `AppAEmployee`. La méthode `MiseAJour` interagit avec l'application de façon à effectuer la mise à jour.

La figure 34 illustre quelques méthodes de gestion des instructions.

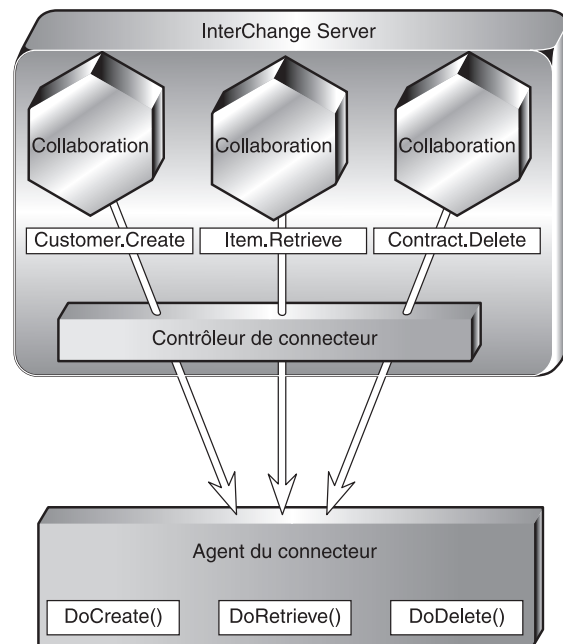


Figure 34. Requêtes de traitement

Lorsque le connecteur décrit en figure 34 reçoit une requête `Customer.Create`, `Item.Retrieve`, ou `Contract.Delete`, il appelle respectivement les méthodes `DoCreate`, `DoRetrieve`, ou `DoDelete`.

informations propres à l'application basées sur des instructions

Dans la définition d'un objet métier, les informations propres à l'application fournissent des éléments complémentaires à l'agent du connecteur qui est en train de traiter une instance de l'objet métier. Chaque instruction peut comporter des informations propres à l'application. Le format et le contenu même des informations propres à l'application sont totalement spécifiques au connecteur.

Par exemple, les informations propres à l'application pour l'instruction `Retrieve` dans la définition d'un objet métier peuvent fournir des arguments d'entrée spéciaux à la méthode `Retrieve` de cet agent du connecteur.

Par exemple, supposez que l'application a trois formulaires dans lesquels des informations à propos d'`InventoryItem` apparaissent :

- `InventoryItem-New`
- `InventoryItem-Change`
- `InventoryItem-Remove`

Lorsque l'agent du connecteur `MyApp` exécute une opération sur un article d'inventaire, il doit faire référence au formulaire correspondant à cette opération. Le champ d'information propre à l'application associé à chaque instruction dans la définition de l'objet métier `InventoryItem` peut stocker le nom du formulaire.

La combinaison des méthodes propres à l'instruction et des entrées propres à l'application fournit à l'agent du connecteur des instructions uniques pour le traitement.

Capacités de traitement simultané

La nature multiprocessus du serveur en java InterChange Server permet un traitement simultané des événements que les contrôleurs de connecteur distribuent aux collaborations.

Selon les applications et les connecteurs mis en oeuvre, le traitement simultané peut également être utilisé pour les requêtes reçues par les agents du connecteur. Ceci est réalisable par l'intermédiaire de plusieurs unités d'exécution ou de l'exécution de traitements en parallèle. Cette capacité peut être utilisée pour des requêtes en provenance de collaborations ou de clients d'accès.

La façon dont le traitement en parallèle peut être mis en oeuvre dépend du type et de la conception des connecteurs :

- Si un agent du connecteur a été écrit en C++, il est par nature monoprocessus. Cependant, les agents du connecteur en C++ peuvent être définis pour exécuter simultanément plusieurs processus en utilisant le parallélisme de l'agent du connecteur, une fonction qui exécute des processus en simultané en créant plusieurs instances de processus esclaves à partir d'un processus maître monoprocessus de l'agent du connecteur. Cette fonction peut être activée ou désactivée à partir des paramètres configurables dans System Manager.
- Si un agent du connecteur a été écrit en Java, il est par nature en mesure d'exécuter des traitements simultanés grâce au traitement multitâche, sans avoir à utiliser la fonction de parallélisme de l'agent du connecteur.
- Certains agents du connecteur écrits en Java peuvent avoir été conçus pour exécuter les traitements en monoprocessus, même si le kit de développement de connecteurs Java (JCDK) est lui-même multiprocessus. Un agent du connecteur Java est généralement conçu de cette manière lorsque les bibliothèques de l'interface de programme d'application de l'application imposent des restrictions selon lesquelles l'agent du connecteur ne doit fonctionner qu'en mode monoprocessus. Pour de tels connecteurs, la fonction de parallélisme de l'agent du connecteur peut être utilisée pour instancier de multiples processus esclaves à partir d'un processus maître de l'agent du connecteur.

Notez bien que l'utilisation du parallélisme de l'agent du connecteur exige que l'application soit elle-même capable de prendre en charge le traitement simultané. Certaines applications peuvent avoir des architectures ou des spécifications relatives aux traitements qui rendent le traitement simultané peu pratique.

Construction et déconstruction des objets métier

Un agent du connecteur réalise ses notifications d'événements et ses tâches de gestion des requêtes en construisant et déconstruisant des objets métier :

- Lorsqu'un agent du connecteur extrait un événement qu'il doit envoyer à InterChange Server, il construit un objet métier qui représente l'événement.
- Lorsqu'un agent du connecteur reçoit un objet métier qui représente une requête provenant d'une collaboration, il déconstruit l'objet métier pour créer une requête de l'application.

Métadonnées des objets métier et les actions des connecteurs

La transformation d'un événement d'application en objet métier et la transformation d'un objet métier en requête de l'application par un connecteur est conduite par les définitions de données (les **métadonnées**) qui sont définies lors de la conception d'un objet métier.

Les agents du connecteur et les métadonnées des objets métier sont conçus pour fonctionner ensemble. La conception d'un agent du connecteur et de ses objets métier est similaire à la conception d'un périphérique d'ordinateur, pour lequel certaines fonctionnalités peuvent être mises en oeuvre soit par le logiciel, soit par le matériel. Le développeur prend en compte les performances, l'évolutivité ainsi que d'autres facteurs pour décider des modalités d'implémentation des principales fonctions.

D'une façon similaire, la répartition du travail entre l'agent du connecteur et ses objets métier sera décidée par le concepteur lors du développement du connecteur. Les principes de conception d'IBM WebSphere InterChange Server favorisent l'utilisation des métadonnées de l'objet métier pour conduire la logique du connecteur, plutôt que le codage en dur de la logique dans les agents du connecteur.

En plus des propriétés qui définissent le type, la taille et la valeur par défaut des attributs, la définition de l'objet métier utilise des champs propres à l'application pour transmettre des instructions spécifiques à l'agent du connecteur sur la façon de traiter l'objet métier.

Par exemple, souvenez-vous que le Chapitre 4, «Objets métier», à la page 59, présente quelques exemples d'informations propres à l'application pour les attributs de l'objet métier qui représente un client. Le tableau 7 présente quelques-uns de ces exemples.

Tableau 7. Exemple d'informations propres à l'application

Attribut	Informations propres à l'application
ID client	CUST1:CID
Nom du client	CUST1:CNAM
Statut	CUST1:CSTAT

Lors du traitement d'un objet métier, l'agent du connecteur lit les définitions et utilise les informations propres à l'application pour construire une requête de l'application :

- Pour obtenir l'ID client, l'agent du connecteur récupère la valeur du champ CID du formulaire CUST1.
- Pour obtenir le nom du client, l'agent du connecteur récupère la valeur du champ CNAM du formulaire CUST1.
- Pour obtenir le statut du client, l'agent du connecteur récupère la valeur du champ CSTAT du formulaire CUST1.

Étant donné que les informations propres à l'application et autres métadonnées contenues dans la définition de l'objet métier guident les actions de l'agent du connecteur, le comportement de l'agent du connecteur peut être défini comme **basé sur les métadonnées**.

Avantages des agents du connecteur basés sur les métadonnées

Un agent du connecteur basé sur les métadonnées est souple, car aucune de ses instructions sur la façon de gérer chaque type d'objet métier qu'il doit prendre en charge n'est codée en dur. Sans aucune modification du code ni recompilation, l'agent du connecteur prend en charge automatiquement de nouvelles définitions d'objets métier, du moment qu'elles correspondent aux spécifications du connecteur.

Un agent du connecteur basé sur les métadonnées prend également en charge les attributs nouveaux ou modifiés dans la définition de l'objet métier. L'agent du connecteur traite automatiquement les attributs lorsqu'il effectue sa boucle à travers les attributs de la définition de l'objet métier.

Exemple de construction d'objet métier

Le processus suivant décrit la façon dont un agent du connecteur crée un objet métier à partir de sa définition :

1. L'agent du connecteur effectue sa boucle à travers les attributs de définition des objets métier, attribut par attribut, en utilisant les informations propres à l'application pour préparer un appel de l'interface de programme d'application ou construire une requête pour obtenir l'entité d'application.
2. L'agent du connecteur envoie alors la requête à l'application et extrait les résultats.
3. L'agent du connecteur effectue sa boucle à travers les résultats, en utilisant les valeurs d'AppSpecificInfo pour déterminer quelles valeurs extraites représentent chaque attribut d'objet métier.

La figure 35 est un exemple d'agent du connecteur en train de construire un objet métier à partir de sa définition. L'agent du connecteur a extrait un événement d'application impliquant un élément dont la valeur principale, le numéro d'élément, est 123. L'agent du connecteur doit construire un objet métier élément à partir de la définition de l'objet métier, qui contient quatre attributs : Group, Description, Price et ItemNum.

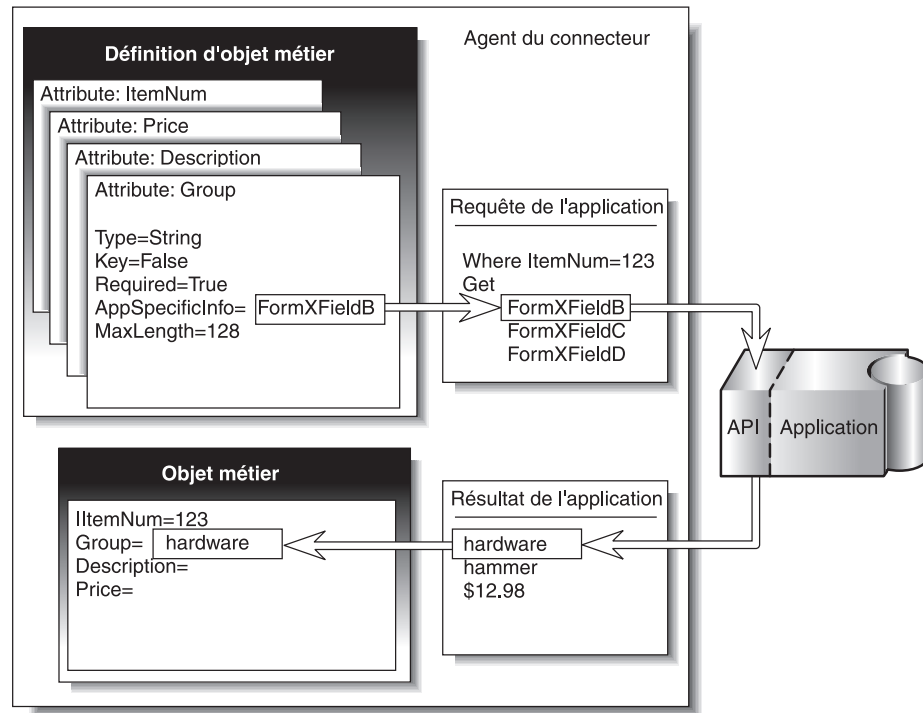


Figure 35. Construction d'un objet métier dans un connecteur

À l'aide du numéro d'élément, 123, pour identifier l'élément, l'agent du connecteur extrait les valeurs des attributs restants. Les informations propres à l'application fournissent l'identifiant du formulaire et du champ pour les données souhaitées.

Par exemple, FormXFieldB identifie les données de l'attribut de groupe. L'agent du connecteur demande la valeur du champ B dans le formulaire X pour l'élément ID 123. L'agent du connecteur utilise alors la valeur retournée "hardware," pour renseigner la valeur de l'attribut de groupe de l'objet métier.

Le processus de déconstruction fonctionne de la façon inverse. L'agent du connecteur utilise la définition de l'objet métier pour déterminer la façon de fabriquer un requête de l'application à partir des données contenues dans l'objet métier qui a été reçu.

Configuration du connecteur

Avant de pouvoir utiliser un connecteur, vous devez lui fournir une définition de connecteur, qui contient les informations suivantes :

- Valeurs des propriétés de la configuration du connecteur qui doivent être utilisées par l'agent du connecteur et le contrôleur de connecteur
- Les objets métier qui seront pris en charge par le connecteur
- Dans certains cas, les mappes spécifiques à utiliser entre des objets métier spécifiés

L'outil Connector Configurator vous fournit l'interface pour définir ces informations.

Propriétés du connecteur

Il existe deux types de propriétés de connecteurs :

- **Propriétés standard**

Les propriétés standard sont disponibles pour *tous* les connecteurs. Les propriétés standard sont les suivantes :

- L'indication que l'agent du connecteur journalise des messages localement ou les envoie au serveur InterChange Server pour les inclure dans le fichier journal général
- L'emplacement du fichier journal pour les agents du connecteur qui journalisent localement
- La connexion et le mot de passe de l'application

- **Propriétés propres aux connecteurs**

Les propriétés propres au connecteur sont en général des valeurs dont un agent du connecteur spécifique a besoin pour établir une session avec l'application. Voici des exemples de propriétés propres aux connecteurs pour différents connecteurs :

- Nom ou adresse IP de la machine qui exécute l'application
- Identification des systèmes de la passerelle d'application
- Nom de la base de données de l'application
- Nom de la boîte de réception des événements

Les propriétés du connecteur peuvent être configurées à l'aide de l'outil Connector Configurator. Hormis quelques restrictions, les propriétés du connecteur peuvent également être définies dans des fichiers de configuration locaux qui résident sur la machine où l'agent du connecteur est installé. Certaines propriétés de configuration du connecteur peuvent également être définies à partir de la ligne de commande.

Mappes associées

Le contrôleur de connecteur utilise des références de mappes lorsqu'il reçoit un objet métier qui nécessite un mappage. Si des transformations de données sont nécessaires entre un objet métier générique particulier et un objet métier propre à l'application, vous devez indiquer la mappe qui va réaliser ces transformations. Vous indiquez les mappes associées dans Connector Configurator.

Développement du connecteur

Lorsque vous développez ou modifiez un connecteur, vous créez l'agent du connecteur lui-même et les définitions des objets métier qu'il va utiliser, puis vous créez une définition du référentiel du connecteur. Notez bien que vous n'avez pas besoin de créer ni de modifier un contrôleur de connecteur, car le contrôleur de connecteur est un composant interne d'InterChange Server et est donc instancié par InterChange Server pour chaque connecteur que vous définissez dans le référentiel.

Le développement d'un connecteur implique la création des relations entre ce connecteur et une application particulière. Le codage en absolu d'un connecteur est généralement un processus relativement simple. Les tâches les plus pointues sont :

- La conception de la méthode de notification d'événement de l'application
- La définition des objets métier propres à une application et leur mappage avec les définitions des objets métier génériques

- La définition des relations entre les objets métier propres à une application et le connecteur

Pour plus d'information sur l'architecture, la modification et le développement des connecteurs, consultez les manuels suivants : *Connector Development Guide for Java* ou *Connector Development Guide for C++*.

Récapitulatif

Ce chapitre vous a présenté les connecteurs et la façon dont ils fonctionnent. Les principaux points à retenir sont :

- Les connecteurs ont deux rôles principaux. Ils doivent notifier aux collaborations les événements de l'application et prendre en charge les requêtes de l'application au nom des collaborations.
- Dans son rôle de notification d'événement, le connecteur interagit avec l'application pour détecter les changements dans l'application et traiter les données associées à ces changements.
- Dans son rôle de mise en oeuvre des requêtes des collaborations, le connecteur utilise des fonctions uniques qui implémentent chaque instruction d'objet métier que le connecteur prend en charge.
- Lorsqu'un agent du connecteur construit un objet métier à partir d'un événement d'une application, ou déconstruit un objet métier pour générer une requête de l'application, l'agent du connecteur se base sur les informations propres à l'application et les métadonnées contenues dans la définition de l'objet métier.

Le prochain chapitre étudie plus en détail le fonctionnement du mappage.

Chapitre 6. Mappage de données

Le **mappage de données** est le processus par lequel le système IBM WebSphere InterChange Server transmet des données à partir d'un objet métier propre à l'application vers un objet métier générique ou à partir d'un objet métier générique vers un objet métier propre à l'application. Le mappage permet au système IBM WebSphere InterChange Server de gérer les différences entre la modélisation des données des différentes applications.

Remarque : Lorsqu'une collaboration transfère des informations à travers plusieurs installations de la *même* application, le mappage n'est *pas* nécessaire à moins que des personnalisations différentes aient été effectuées sur chacune des installations.

Le mappage de données se produit lors de l'exécution dans le système IBM WebSphere InterChange Server, à l'aide des mappes que vous avez créées avant l'exécution.

Ce chapitre présente les processus de mappage et les composants que vous utilisez pour voir, modifier et créer les mappes et les relations. Les sections suivantes seront abordées :

- «Comment le système InterChange Server utilise le mappage»
- «Composants et outils des mappes» à la page 87
- «Transformations du mappage» à la page 88
- «Configuration des connecteurs avec des mappes» à la page 90
- «Récapitulatif» à la page 90

Pour obtenir des instructions détaillées sur l'utilisation de l'outil de mappage d'IBM WebSphere InterChange Server pour créer des mappes, référez-vous au *Map Development Guide*.

Comment le système InterChange Server utilise le mappage

Le mappage permet aux collaborations de prendre les données d'un objet métier d'une application et de les transformer pour générer un objet métier pour une application hétérogène. Lors du processus de mappage, les collaborations interagissent avec les connecteurs, avec l'interface d'accès, ou avec les deux.

Le système IBM WebSphere InterChange Server fournit une prise en charge étendue pour le mappage de données entre les objets métier, y compris les capacités suivantes :

- Transformation des valeurs des données d'un ou plusieurs attributs dans un objet métier source vers un ou plusieurs attributs dans un objet métier cible
- Établissement et maintenance des relations entre des entités de données qui sont équivalentes mais représentées différemment et impossibles à transformer directement
- Accès à des ressources de mappage externes, telles que des bases de données pour établir des interrogations et des produits de mappage tiers

Dans un environnement IBM WebSphere InterChange Server, le mappage se produit généralement entre des objets métier propres à une application et des

objets métier génériques. Le système IBM WebSphere InterChange Server ne mappe pas les objets métier propres à une application directement à d'autres objets métier propres à une application. Au lieu de cela, l'objet générique agit en tant qu'intermédiaire entre deux modèles de données d'applications, en transmettant les informations mappées d'un modèle de données à la collaboration (soit via un connecteur soit via l'interface d'accès au serveur), puis en transférant les informations mappées de la collaboration vers le connecteur d'un autre modèle de données.

La figure 5-1 illustre la façon dont le mappage de données se produit en contexte d'exécution, en utilisant une collaboration de gestion d'employés comme exemple fictif : la collaboration Employee Management reçoit un objet métier Employeee de la part du connecteur source, puis envoie un objet métier Employee au connecteur cible. (Dans cet exemple, une collaboration reçoit un objet métier d'un connecteur : un processus de mappage similaire se produit lorsqu'une collaboration reçoit un objet métier de l'interface d'accès).

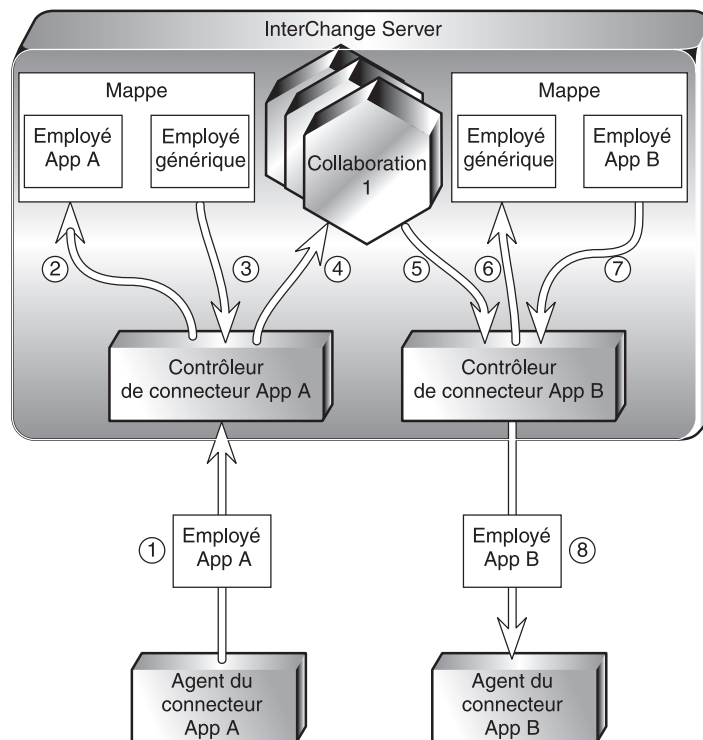


Figure 36. Mappage de données en contexte d'exécution

La figure 5-1 illustre la séquence suivante :

1. L'agent du connecteur App A produit un objet métier App A Employee et l'envoie au contrôleur de connecteur.
2. Le contrôleur de connecteur transmet l'objet métier App A Employee à InterChange Server pour le mappage. La requête inclut le nom de la mappe de données que le serveur doit utiliser, sur la base du nom de mappe indiqué dans la configuration du connecteur.
3. La mappe retourne l'objet métier générique Employee au contrôleur de connecteur.

4. Le contrôleur de connecteur vérifie les collaborations qui sont abonnées à l'objet métier générique Employee. Dans le cas présent, la Collaboration1 a un abonnement, le contrôleur de connecteur transmet donc l'objet métier à la Collaboration1.
5. La collaboration effectue quelques traitements puis produit un autre objet métier générique Employee en sortie, qu'elle envoie au contrôleur de connecteur.
6. Le contrôleur de connecteur transmet l'objet métier générique à InterChange Server, en demandant un mappage à destination de l'objet métier App B Employee.
7. La mappe retourne l'objet métier propre à l'application au contrôleur de connecteur.
8. Le contrôleur de connecteur transmet l'objet métier App B à l'agent du connecteur App B, qui peut alors transmettre les données de l'objet métier à l'Application B.

L'exemple ci-dessus a utilisé deux mappes : une entre l'objet métier App A Employee et l'objet métier générique Employee utilisé par la collaboration, et une autre entre l'objet métier générique Employee et l'objet métier App B Employee. Les données Employee n'ont été transférées que dans une seule direction : de App A vers App B.

Si vous aviez voulu échanger les données Employee dans les *deux* directions entre les deux différentes applications, quatre mappes auraient été nécessaires :

- Une mappe de l'objet métier propre à l'application de l'Application A vers l'objet métier générique utilisé par la collaboration.
- Une mappe de l'objet métier générique vers l'objet métier propre à l'application de l'Application B.
- Une mappe de l'objet métier propre à l'application de l'Application B vers l'objet métier générique.
- Une mappe de l'objet métier générique vers l'objet métier propre à l'application de l'Application A.

Composants et outils des mappes

Le système IBM WebSphere InterChange Server contient une interface de programme d'application de mappage Java, l' API de mappage, qui inclut des méthodes pour gérer les situations de transformations de données communes. Les outils de conception graphique sont fournis pour créer les deux composants principaux d'IBM InterChange Server pour le mappage, les définitions des mappes et des relations :

- Une **mappe** contient le code Java qui indique la façon de transformer les attributs d'un ou plusieurs objets métier source à destination d'un ou plusieurs objets métier cible. Une mappe est nécessaire pour chaque objet métier que vous souhaitez transférer entre différentes applications. Lorsque vous modifiez des objets métier, vous devez également modifier les mappes correspondantes. En général, une mappe est créée pour chaque objet métier source à transformer. Le logiciel **Map Designer** est utilisé pour créer et compiler les mappes.
- Une **définition de relation** établit une association entre deux entités de données ou plus dans le système IBM WebSphere InterChange Server. Les définitions de relations à l'intérieur des mappes sont souvent utilisées pour transformer les attributs des objets métier dont les données ont des fonctions similaires mais une

représentation différente dans les différentes applications. La plupart des mappes utilisent une ou plusieurs définitions.

L'outil **Relationship Designer** sert à créer des définitions de relations ainsi que les schémas des tables qui sont utilisés pour stocker les données en instance de relation lors de l'exécution.

Les définitions des mappes et des relations sont toutes deux stockées dans le référentiel d'InterChange Server. De même que pour les définitions d'objets métier, les définitions de relations fonctionnent comme des spécifications ou des modèles pour les instances qui sont créées. Contrairement aux instances des objets métier, les instances des relations sont persistantes, et chaque relation est stockée dans une table spéciale.

À chaque fois que le système reçoit une requête pour transformer un objet métier, il exécute la mappe associée et, selon l'objectif de la transformation, crée une ou plusieurs instances des définitions de relations associées. Les instances des relations créées lors de l'exécution des mappes contiennent les données de l'exécution des attributs auxquels elles sont associées, et ces données sont stockées dans les tables des relations.

Pour des informations complémentaires sur le fonctionnement de l'outil de mappage, référez-vous au chapitre 1 du manuel *Map Development Guide*.

Transformations du mappage

Une mappe associe un objet métier source avec un objet métier cible, et contient un ensemble d'étapes de transformation, une pour chaque attribut qui est transformé. Chaque étape de transformation contient du code Java qui calcule la valeur de l'attribut.

Dans une mappe de données, la conversion des attributs source vers les attributs cible peut se dérouler simplement, ou bien exiger l'établissement et la maintenance de relations entre les entités de données qui sont équivalentes mais représentées différemment et impossibles à transformer directement.

Transformations simples

Dans une configuration simple de transformation de données, les valeurs des attributs source et cible ont une corrélation claire et des significations identiques, même si les attributs sont structurés différemment.

Le mappage simple peut comprendre les opérations suivantes :

- Copie d'une valeur d'attribut source vers une ou plusieurs valeurs d'attributs cible.
- Division d'une valeur d'attribut source en plusieurs valeurs d'attributs cible.
- Jonction de plusieurs valeurs d'attributs source en une seule valeur d'attribut cible.
- Ignorer une valeur d'attribut source pour lequel l'objet métier cible n'a pas d'attribut équivalent.

La figure 5-2 montre un exemple de mappage simple pour certaines de ces opérations :

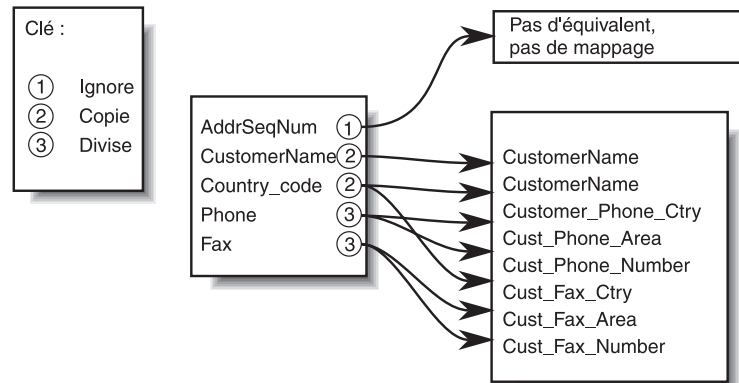


Figure 37. Le mappage simple

Transformations des relations

Certains attributs ne se prêtent pas à de simples transformations. Certaines applications différentes peuvent avoir des attributs similaires, contenant des informations dont la fonction est équivalente, mais dont les formats et les valeurs ne sont pas compatibles. Par exemple, pour un attribut de pays, une application peut utiliser un code à deux lettres (tels US, FR, ou EG) tandis qu'une autre application utilise un code numéral (tels 1, 2, ou 3).

Pour associer de tels attributs entre différentes applications, vous devez créer des **définitions de relations**, associant les données des attributs source et cible.

Catégories de relations

Lorsque vous créez une définition de relation, vous devez établir une liste des participants (objets métier et autres entités de données) qui sont impliqués dans la relation, et ensuite spécifier un type (soit le nom de la définition de l'objet métier, soit le mot représentant les données) pour chaque participant. Selon le type des participants et le nombre d'instances de chaque participant pouvant être associées à cette définition, les relations sont classées dans les catégories suivantes :

- Une **relation d'identité** établit une association entre des objets métier et d'autres données sur la base d'un rapport un à un. Pour chaque instance d'une relation il ne peut y avoir qu'une instance de chaque participant. Les relations d'identité transforment généralement les attributs principaux des objets métier, tels le numéro d'identificateur et le code produit.
- Une **relation de non-identité** établit une association entre des objets métier ou d'autres données sur une base un à un. Chaque instance de relation peut comporter une ou plusieurs instances de chaque participant. Comme exemple de relation de non-identité, nous avons la transformation RMA-vers-Commande, dans laquelle un objet métier RMA (autorisation de retour de matériaux) simple peut renvoyer un ou plusieurs objets métier Commande.
- Une **relation de recherche** établit une association entre des données, telles que des attributs dans des objets métier. Les données peuvent être associées selon une base un à un, un à plusieurs ou plusieurs à plusieurs. Les participants à la relation de recherche sont du type Données. Les relations de recherche transforment généralement des attributs secondaires dont les valeurs sont représentées par des codes, tels que la situation de famille ou le code devise.

Pour des informations complémentaires sur les définitions des relations, référez-vous au chapitre 1 du manuel *Map Development Guide*.

Configuration des connecteurs avec des mappes

Lorsque vous utilisez System Manager pour installer et configurer un connecteur, vous devez indiquer le nom d'une mappe d'IBM WebSphere InterChange Server, qui est stockée dans le référentiel d'InterChange Server, afin de pouvoir transformer chacun des objets métier que le connecteur prend en charge.

Lorsqu'un contrôleur de connecteur envoie un objet métier pour un mappage, il exécute la mappe en envoyant l'objet métier et les informations relatives à la mappe en tant qu'entrées.

Récapitulatif

Ce chapitre vous a présenté la terminologie et les concepts associés au mappage des objets métier. Les principaux points à retenir sont :

- Le mappage de données transforme des objets métier. Étant donné que les objets métier voyagent d'une application source vers des collaborations, le mappage transforme des objets métier propres à une application en objets métier génériques. Étant donné que les objets métier voyagent depuis des collaborations vers des applications cible, le mappage transforme des objets métier génériques en objets métier propres à une application.
- Les mappes peuvent être créées avec les outils Map Designer et Relationship Designer.
- Le mappage simple peut être réalisé dans Map Designer. Pour les mappages qui impliquent des attributs qui sont équivalents mais dont le format est incompatible, utilisez Relationship Designer.

Tous les chapitres précédents décrivent les fonctions de base du système IBM WebSphere InterChange Server. Le chapitre suivant décrit une fonction évoluée destinée à quelques utilisateurs.

Chapitre 7. Collaborations transactionnelles

Les collaborations transactionnelles fournissent des garanties de cohérence dans les processus métier. Ce chapitre présente les fonctions des collaborations transactionnelles. Les sections suivantes seront abordées :

- «Modèle de transaction»
- «Qu'est-ce que la collaboration transactionnelle ?» à la page 92
- «Isolement des données» à la page 97
- «Niveaux de transaction» à la page 98
- «Reprise sur incident» à la page 100
- «Collaborations transactionnelles et les processus métier longue durée» à la page 101
- «Récapitulatif» à la page 101

Modèle de transaction

Cette section passe brièvement en revue les concepts dont sont issus les modèles transactionnels d'IBM WebSphere InterChange Server. Si les bases de données et le traitement transactionnel vous sont familiers, passez à la section suivante.

Une transaction est un ensemble d'étapes associées et interdépendantes formant une unité logique de travail. Le regroupement des étapes dans une transaction garantit que l'ensemble des étapes sont traitées en tant qu'élément unique : soit l'ensemble complet d'étapes réussit, soit l'ensemble complet d'étape échoue.

La figure 38 illustre la transaction d'une base de données contenant deux mises à jour, mise à jour 1 et mise à jour 2.

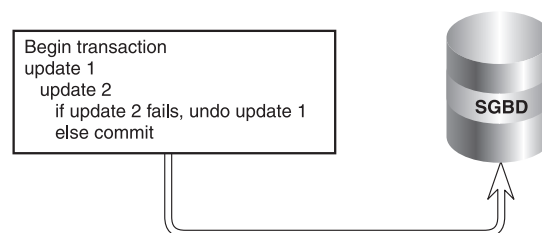


Figure 38. Une transaction de base de données

Lorsque la transaction est en cours d'exécution, elle verrouille la partie de la base de données qu'elle est en train de modifier, rendant les données inaccessibles à la modification par d'autres transactions. L'isolement des données des interférences en provenance de l'extérieur fait partie des qualités de définition des transactions.

Si les opérations de la transaction réussissent, la transaction s'achève en écrivant les modifications sur le disque par une opération de **validation**. Une opération de validation libère les verrous de la transaction, rendant ainsi les données mises à jour disponibles pour les autres transactions.

Si une erreur survient en cours d'exécution, rendant ainsi l'ensemble de la transaction impossible, la transaction échoue. Plutôt que de laisser des résultats

partiels dans la base de données, la transaction annule les changements qu'elle a déjà effectués, laissant ainsi la base de données avec les mêmes valeurs qu'avant le début de la transaction. Ce processus d'annulation est appelé **invalidation**.

Un autre type de transaction de base de données, la validation en deux phases, convient à une utilisation répartie entre de multiples bases de données. La validation en deux phases utilise un moniteur de traitement transactionnel pour coordonner les mises à jour concurrentes. Le moniteur de traitement transactionnel vérifie d'abord que toutes les bases de données peuvent effectuer la modification souhaitée. Même si elle le peuvent toutes, elle doivent attendre le signal du moniteur de traitement transactionnel pour effectuer la modification. Si toutes ne peuvent pas effectuer la modification, aucune ne le fait.

Chaque transaction doit maintenir la cohérence dans sa base de données, mais le protocole de validation à deux phases étend la portée de cette condition au-delà des bases de données individuelles. L'exemple typique est le transfert de fonds, dans lequel le moniteur de traitement transactionnel s'assure que les fonds débités sur un compte sont crédités sur un autre compte, sinon aucun des deux comptes n'est modifié.

Qu'est-ce que la collaboration transactionnelle ?

Une collaboration transactionnelle est une collaboration qui peut être annulée. De même que pour les transactions de bases de données, les collaborations transactionnelles sont des opérations tout-ou-rien : soit l'intégralité de la collaboration réussit, soit l'intégralité de la collaboration échoue. De plus, une collaboration transactionnelle peut détecter toute violation de l'isolement de ses données qui pourraient compromettre la logique de ses opérations sur les données, et elle peut réagir contre cette violation.

Les collaborations transactionnelles sont fondées sur les principes établis dans les transactions des bases de données et les protocoles de validation à deux phases. Cependant, les collaborations transactionnelles diffèrent des transactions des bases de données à cause de la nature particulière des collaborations : réparties entre un nombre illimité d'applications, asynchrones, longue durée, et placées à l'extérieur des applications plutôt qu'à l'intérieur.

Une collaboration transactionnelle s'exécute différemment d'une collaboration non transactionnelle. Une collaboration transactionnelle s'exécute sous le contrôle du service de transaction d'InterChange Server, qui gère l'exécution, l'invalidation et vérifie l'isolement.

Cette section décrit les composants des collaborations transactionnelles.

Scénarios transactionnels

La propriété du niveau de transaction minimum d'un modèle de collaboration détermine si la collaboration est transactionnelle, et s'applique à tous ses scénarios. Tous les objets de collaboration créés à partir d'un modèle héritent de son niveau de transaction minimum.

La logique transactionnelle en cours s'applique au niveau du scénario. Dans une collaboration transactionnelle, chaque scénario est un **scénario transactionnel**, dont le démarrage commence implicitement une transaction et dont l'achèvement réussi valide implicitement la transaction.

Chaque scénario s'exécute dans un contexte d'exécution simple. Lorsqu'une collaboration fait partie d'un groupe de collaborations, un scénario peut appeler une autre collaboration pour effectuer une tâche puis renvoyer le résultat. Le scénario appelé s'exécute dans le contexte d'exécution de l'appelant et fait partie de la même transaction que ce dernier.

Sous-transactions

À chaque fois qu'une étape d'un scénario entraîne la modification de données par une application, cette étape initie une transaction dans l'application elle-même. Une étape qui entraîne une transaction dans une application s'appelle une **étape de sous-transaction** : une transaction à l'intérieur d'une transaction, où la transaction principale est le scénario lui-même.

Pour illustrer les concepts transactionnels, ce chapitre utilise le scénario développé dans la figure 39. Ce scénario gère les mises à jour formant partie d'une collaboration de gestion des tarifications. La collaboration garantit que lorsqu'une application ERP modifie le prix d'un produit, la modification est répercutée dans l'application du service clientèle et dans l'application de configuration. Cette illustration omet tous les détails du traitement qui ne sont pas pertinents dans la sémantique transactionnelle.

Le scénario est constitué de quatre étapes d'action, Action 1 (A1) à Action 4 (A4), et il interagit avec trois objets métier : A, B et C. L'objet métier A représente un produit particulier dans le système ERP, le B représente le même produit dans l'application du service clientèle et le C représente le produit dans l'application de configuration du produit.

Remarque : Le format *ObjetMétier.Attribut* représente un attribut particulier dans un objet métier. Par exemple, *A.Prix* représente l'attribut Prix de l'objet métier A.

Le scénario apparaît sur la gauche. Sur la droite, les commentaires indiquent ce qui se passe lorsqu'un responsable produit se connecte au système ERP et modifie le prix du produit à 700 \$.

Remarque : Pour plus de simplicité, les connecteurs ne figurent pas dans les schémas de ce chapitre.

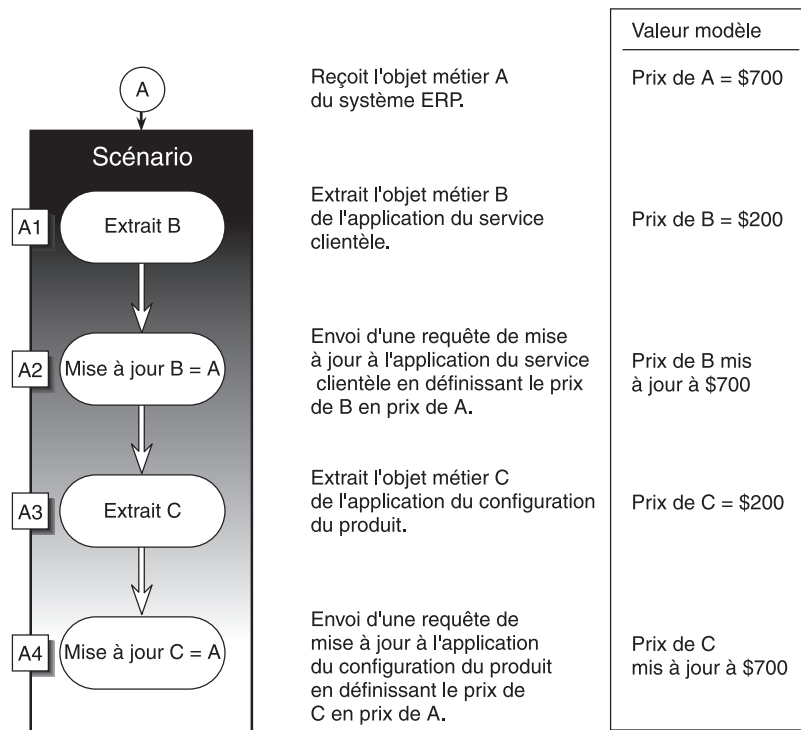


Figure 39. Modèle de scénario

Pour que le scénario soit transactionnel, la première étape consiste à identifier les étapes sous-transactionnelles. La figure 40 montre que les étapes sous-transactionnelles du scénario sont les étapes A2 et A4, car elles génèrent des transactions dans les applications du service clientèle et de configuration du produit, respectivement.

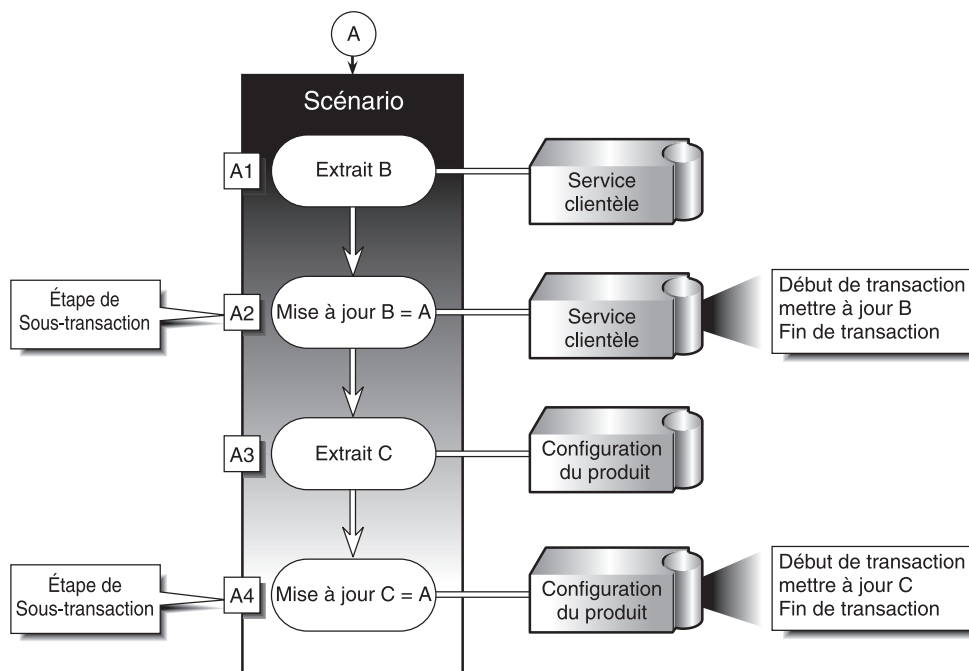


Figure 40. Étapes des sous-transactions dans un scénario transactionnel

Une étape qui effectue une requête d'extraction, telles A1 ou A3, n'est pas une étape sous-transactionnelle, car elle n'entraîne pas de modification des données.

Correction et l'invalidation

La façon dont un scénario transactionnel réagit à un incident est différente de celle des scénarios non transactionnels. Lorsqu'un scénario non transactionnel échoue, la collaboration enregistre simplement l'erreur dans un journal puis s'achève. Lorsqu'un scénario transactionnel échoue, le scénario invalide son action pour maintenir la cohérence des données dans les différentes bases de données impliquées.

Utilisation de la correction lors de l'invalidation

Lorsqu'une erreur se produit, les sous-transactions peuvent déjà avoir demandé aux applications de valider le travail. Par conséquent, l'invalidation s'effectue sous la forme d'**étapes de correction** qui annulent les effets des autres actions. Une étape de correction n'a lieu que lors d'une procédure d'invalidation.

Pendant l'invalidation, InterChange Server suit les étapes selon l'ordre inverse du chemin d'exécution. Le serveur exécute les corrections correspondant à chaque étape de sous-transaction terminée. L'invalidation est achevée lorsque toutes les étapes transactionnelles exécutées ont été corrigées. Si une erreur survient lors du processus d'invalidation, InterChange Server enregistre simplement l'erreur dans un journal d'erreurs.

Une étape de correction est constituée de toute action que le développeur de collaborations souhaite utiliser pour annuler l'action d'origine. Le tableau 8 liste plusieurs façons habituelles que les développeurs de collaborations peuvent choisir pour corriger des actions spécifiques.

Tableau 8. Exemples de corrections

Action	Correction
Créer un objet métier	Supprimer un objet métier
Supprimer un objet métier	Créer un objet métier
Mettre à jour un objet métier	Mettre à jour un objet métier, en restaurant les valeurs précédentes

Bien que la correction consiste généralement à inverser les modifications de données des actions d'origine, cette inversion n'est pas nécessaire. Par exemple, la correction d'une demande de création pourrait être une autre demande de création, entraînant cette fois un enregistrement dans un journal de suivi. La correction est par conséquent une opération d'**annulation logique** et ne requiert pas nécessairement une opération d'annulation réelle.

Conception d'un processus d'invalidation

Un concepteur de collaborations doit concevoir le scénario permettant à l'invalidation de se produire. La figure 41 montre la façon dont l'exemple de scénario pourrait être modifié pour permettre une invalidation.

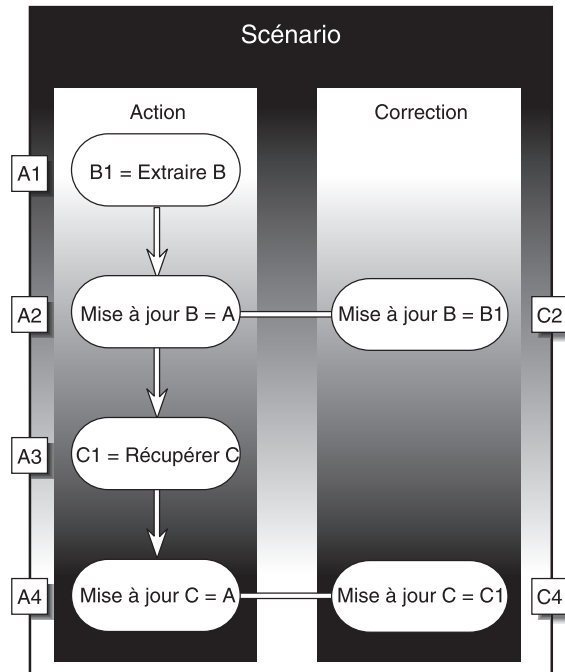


Figure 41. Actions et corrections

L'exemple a été modifié de la façon suivante :

- Les étapes d'action A1 et A3 utilisent désormais B^1 et C^1 pour stocker les valeurs d'origine de B et C.
- Les étapes de correction C2 et C4 représentent les nouvelles étapes de correction pour restaurer les valeurs d'origine de B et C

Imaginez qu'une erreur se produise dans le scénario après l'exécution de l'étape d'action A2. Pendant la phase d'invalidation, l'étape de correction C2 s'exécute en utilisant la valeur stockée en B^1 pour rendre à B sa valeur d'origine.

Illustration de la phase d'exécution

La figure suivante illustre l'exécution d'un scénario modèle. Dans la séquence d'exécution illustrée, les étapes d'action A1, A2 et A3 s'exécutent avec succès. Pendant l'exécution de l'étape A4, cependant, une erreur se produit dans l'application. Lorsque le scénario reçoit un statut d'erreur à la place d'un statut de succès pour l'action A4, il échoue, et la phase d'invalidation commence.

Lors du retour pas à pas à travers les différentes étapes, le service de transaction procède de la façon suivante :

- A4 n'a jamais effectué la modification des données, aucune correction n'est donc nécessaire.
- A3 était une opération d'extraction, aucune correction n'est donc nécessaire.

- A2 a réalisé une modification de données et sa correction, définie en C2, est par conséquent exécutée.

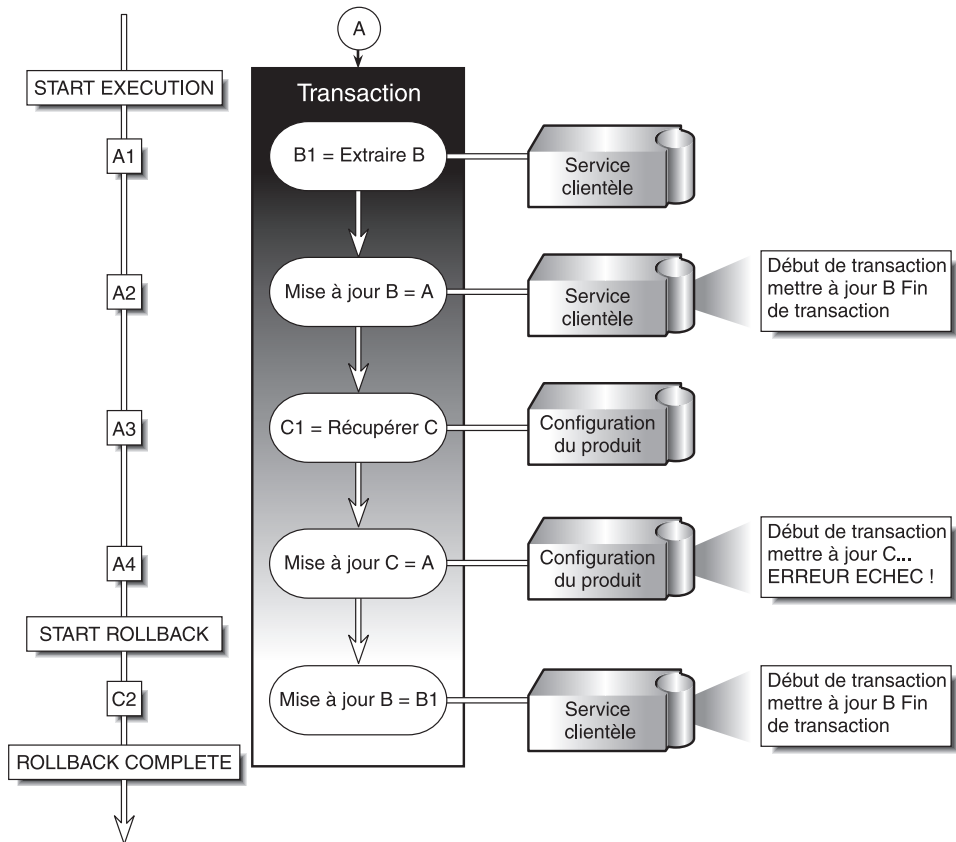


Figure 42. Invalidation d'un scénario transactionnel

Isolement des données

Le système IBM WebSphere InterChange Server fournit plusieurs niveaux d'exécution transactionnelle pour les collaborations. Aux niveaux les plus élevés, InterChange Server détecte la violation des règles d'isolement. L'isolement est la garantie, lors de l'exécution d'une transaction, que celle-ci dispose de l'accès exclusif aux données. Étant donné que les données de la transaction ne changent pas entre les différentes opérations qui se déroulent au sein de cette transaction, les effets de la logique de la transaction sont prévisibles.

Lors d'une transaction de la base de données ou d'une validation en deux phases, les verrous de la base de données garantissent l'isolement pendant l'exécution de la transaction. Dans l'environnement IBM WebSphere InterChange Server, un scénario transactionnel ne peut pas verrouiller les données entre les étapes : chaque étape génère une transaction de l'application qui, une fois terminée, enlève son verrou. D'autres programmes peuvent alors voir et modifier les données mises à jour.

Un scénario peut avoir besoin d'utiliser la même information plus d'une fois. Par exemple, plusieurs étapes d'action peuvent mettre à jour les mêmes données. De plus, si un scénario échoue et procède à une invalidation, l'une des étapes de la correction pourrait accéder de nouveau aux données définies par des actions préalables, afin de restaurer leur valeur d'origine.

En ce qui concerne les données que le scénario utilise plus d'une fois, le délai entre l'achèvement d'une transaction de l'application causé par une étape et le démarrage d'une transaction de l'application causé par une autre étape représente une **fenêtre de vulnérabilité**.

La figure 43 illustre la fenêtre de vulnérabilité entre deux étapes transactionnelles qui mettent à jour le même objet.

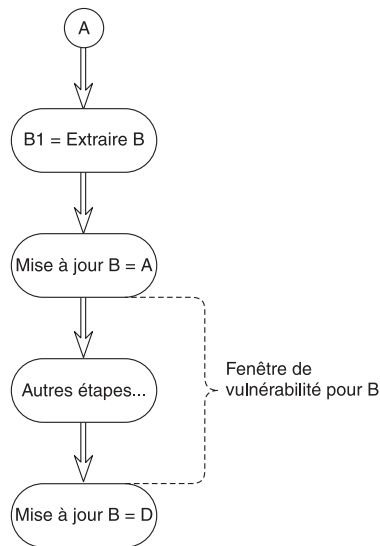


Figure 43. Fenêtre de vulnérabilité

Pendant la fenêtre de vulnérabilité, l'isolement des données ne peut pas être garanti. D'autres transactions pourraient accéder aux données pour changer l'état des données entre deux accès du scénario, pouvant ainsi générer des erreurs dans les résultats du scénario.

Pour certaines collaborations, l'isolement peut se révéler inutile, par exemple lorsque la collaboration est le seul programme à modifier les données. Cependant, si un scénario modifie des données auxquelles d'autres programmes accèdent en modification, les violations de l'isolement peuvent être problématiques.

Le système IBM WebSphere InterChange Server fournit une **vérification d'isolement** pour les collaborations qui requièrent une protection contre les violations d'isolement. Lorsqu'une collaboration exécute le niveau de transaction maximal ou le niveau strict, le service de transaction et les connecteurs travaillent ensemble pour déterminer si les données ont été modifiées entre les différents accès. La prochaine section décrit les différents niveaux de transaction et explique la façon dont se déroule la vérification de l'isolement à chaque niveau.

Niveaux de transaction

Le niveau de transaction d'une collaboration détermine les mécanismes selon lesquels le système exécute le scénario dans la collaboration.

Une collaboration devient transactionnelle lors de sa phase de développement, lorsque le développeur détermine si la collaboration requiert une exécution transactionnelle. Si tel est le cas, le développeur ajoute des étapes de correction aux scénarios dans le modèle et indique le niveau de transaction de la collaboration. Chaque collaboration a l'un des niveaux de transaction suivants :

- Aucun
- Niveau minimal
- Niveau maximal (notez bien que ce niveau n'est pas pris en charge pour les collaboration longue durée)
- Strict

Pendant la configuration, System Manager affiche le niveau de transaction minimal d'un modèle de collaboration, dont les objets de collaboration réalisés à partir de ce modèle hériteront.

Cependant, d'autres facteurs influent sur le niveau de transaction auquel l'objet de collaboration va réellement s'exécuter. Lors de la configuration, l'administrateur lie les objets de collaboration aux connecteurs ou à d'autres objets de collaboration, ou bien aux deux. Chaque connecteur ou objet de collaboration dispose d'un niveau de transaction particulier, mais la collaboration doit s'exécuter à un niveau que toutes les liaisons prennent en charge. Par exemple, le niveau de transaction d'un connecteur représente le niveau de prise en charge qu'il peut fournir, d'après les possibilités de son application. La collaboration ne peut pas s'exécuter à un niveau que ses connecteurs ne prennent pas en charge.

Aucun

Une collaboration dont le niveau de transaction est Aucun n'est pas transactionnelle. Si une erreur survient pendant l'exécution de la collaboration, un message d'erreur est enregistré sur le journal.

Même si le modèle de collaboration contient des étapes de correction, l'objet de collaboration créé à partir de ce modèle ne peut pas les exécuter avec un niveau de transaction à Aucun. Ceci pourrait arriver si, par exemple, les liens de l'objet connecteur ne prennent pas en charge un niveau de transaction plus élevé. Si cela arrive, les corrections sont ignorées.

Niveau minimal

Une collaboration dont le niveau de transaction est minimal bénéficie des corrections définies dans les étapes transactionnelles de son scénario. Si une erreur survient lors de l'exécution de son scénario, InterChange Server invalide le scénario, en exécutant une correction pour chaque étape transactionnelle qui s'est exécutée.

Si le scénario a appelé un autre objet de collaboration pour effectuer une partie du travail (cela signifie que la collaboration fait partie d'un groupe de collaborations), le scénario qui s'est exécuté dans cette collaboration s'invalide également.

Le niveau de transaction minimal est adapté aux collaborations dans les circonstances suivantes :

- Il est important d'invalider en cas d'échec pour annuler le travail validé.
- Les scénarios de la collaboration modifient des données qu'aucun autre programme ou collaboration ne modifie, il n'est donc pas nécessaire de recourir à la vérification d'isolement.

Niveau maximal

Une collaboration qui s'exécute avec le niveau de transaction maximal utilise la vérification d'isolement. Le niveau de transaction maximal est adapté aux collaborations qui doivent se protéger contre toute incohérence des données.

Lorsqu'une collaboration s'exécute avec la vérification d'isolement :

- Le service de transaction InterChange Server travaille avec les connecteurs pour vérifier les données auxquelles le scénario accède de nouveau au cours de son exécution.
- Le système vérifie l'état en cours par rapport au dernier état connu avant de procéder à l'opération suivante.
 - Si les données sont les mêmes que lors de la vérification précédente, les données ont été isolées virtuellement, et l'opération se poursuit.
 - Si les données ne sont pas les mêmes que lors de la vérification précédente, le résultat de l'opération est une erreur.

La vérification de l'isolement fournit une bonne sémantique transactionnelle, mais requiert des accès supplémentaires à la base de données InterChange Server et davantage de communications entre les connecteurs et InterChange Server. Ce travail supplémentaire du système se répercute sur les performances. C'est pour cette raison qu'il est important de mesurer l'impact des performances par rapport aux problèmes de cohérence au moment du choix du niveau de transaction.

Cependant, même le niveau de transaction maximal laisse la place à une petite fenêtre de vulnérabilité. Entre le début de la vérification de l'isolement et le moment où la collaboration continue son opération, une autre transaction pourrait modifier les données. Le niveau maximal est le niveau de transaction le plus élevé disponible lorsqu'il existe un grand nombre de combinaisons d'applications, car peu d'API supportent le niveau Strict.

Remarque : Le niveau maximal de transaction n'est pas pris en charge pour les collaborations longue durée.

Strict

Le niveau de transaction Strict fournit le niveau d'isolement le plus élevé, mais il n'est disponible qu'avec les applications dont les API permettent aux applications clientes d'effectuer des tests d'atomicité et de définir des opérations. Le connecteur travaillant avec ce type d'application verrouille les données pendant la vérification de l'isolement, empêchant ainsi un autre programme de modifier les données du scénario. Peu d'applications fournissent actuellement cette possibilité.

Reprise sur incident

Tout programme logiciel court le risque d'être interrompu par une défaillance matérielle ou logicielle qui stoppe de façon inattendue son exécution. InterChange Server dispose d'un mécanisme robuste pour récupérer les transactions en cours d'exécution lorsqu'un événement inattendu interrompt le traitement.

Lorsqu'InterChange Server redémarre après un arrêt inattendu, il vérifie quelle collaborations étaient en cours de transaction au moment de l'interruption. Une reprise sur incident en deux phases commence alors :

1. InterChange Server réactive chaque collaboration transactionnelle interrompue et l'invalidé (sauf si la collaboration était en train d'utiliser la fonction de processus métier longue durée). Pendant cette période, le serveur n'envoie aucun nouvel événement à la collaboration.
2. InterChange Server extrait l'événement déclencheur d'origine pour chaque collaboration interrompue à partir du service de gestion des événements et le redistribue. La collaboration s'exécute, traitant l'événement de nouveau.

Lorsque la reprise sur incident est terminée, InterChange Server autorise la collaboration à traiter de nouveaux événements.

Collaborations transactionnelles et les processus métier longue durée

Les collaborations transactionnelles peuvent utiliser la fonction de processus métier longue durée, qui est disponible en tant qu'option lorsque vous créez le modèle de collaboration. Si un objet de collaboration a été créé avec un modèle dont cette fonction est activée, et qu'une panne d'InterChange Server interrompt un flux en attente de réponse d'une intervention sur appel, ce flux pourra reprendre à partir du contexte sauvegardé pendant la reprise. Mais si cette fonction n'est pas activée, une panne d'InterChange Server déclenche les invalidations de correction.

Récapitulatif

Ce chapitre vous a présenté les collaborations transactionnelles. Les principaux points à retenir sont :

- Le modèle transactionnel d'IBM WebSphere InterChange Server est dérivé de l'utilisation des transactions dans les environnements des systèmes de gestion de base de données. Cependant, le modèle IBM WebSphere InterChange Server est conçu pour s'adapter à la nature unique des collaborations.
- Une collaboration transactionnelle est une collaboration dans laquelle les scénarios sont définis pour gérer l'invalidation, et pour s'exécuter sous le contrôle du service de transaction d'InterChange Server.
- Lorsqu'un scénario échoue dans une collaboration transactionnelle, InterChange Server invalide ce scénario, en exécutant la correction pour chaque action terminée.
- Avec des niveaux de transaction plus élevés, le système peut également fournir la vérification de l'isolement.
- Les mécanismes de reprise sur incident garantissent aux collaborations qu'InterChange Server ne les laissera pas dans un état transactionnel inconnu, même en cas de panne du système ou de tout autre événement inattendu.

Chapitre 8. Prise en charge des comportements spécifiques aux langues

WebSphere InterChange Server et les adaptateurs WebSphere Business Integration (produits WebSphere Business Integration) prennent en charge les langues bidirectionnelles comme l'arabe et l'hébreu via un système appelé *prise en charge des langues bidirectionnelles*. La prise en charge des langues bidirectionnelles est un mécanisme qui permet d'afficher et de traiter de manière exacte les données des scripts bidirectionnels contenus dans les composants (connecteurs, structures d'adaptateurs, collaborations et mappes) intégrés aux produits WebSphere Business Integration. Les produits WebSphere Business Integration peuvent traiter les données dans n'importe quel format bidirectionnel. Toutefois les données du domaine des produits WebSphere Business Integration sont définies dans un unique format bidirectionnel qui est le format standard Windows (logique de gauche à droite).

Ce chapitre donne une présentation générale de la prise en charge des données bidirectionnelles dans les produits WebSphere Business Integration et des conditions et traitements applicables à cette prise en charge. Les sections suivantes seront abordées :

- «Prise en charge de l'environnement local dans les produits WebSphere Business Integration»
- «Prise en charge des données bidirectionnelles» à la page 106
- «Attributs et transformations de la présentation» à la page 110
- «Prise en charge des scripts bidirectionnels dans les produits WebSphere Business Integration» à la page 112
- «limitations conceptuelles» à la page 116

Prise en charge de l'environnement local dans les produits WebSphere Business Integration

L'environnement *local* de l'utilisateur est la partie de l'environnement informatique qui regroupe les informations relatives à la manière de traiter les données spécifiques au pays, à la langue ou à la zone géographique de l'utilisateur. L'environnement local est habituellement installé en même temps que le système d'exploitation.

L'environnement local de l'utilisateur comprend généralement les informations suivantes :

- Les conventions culturelles propres à la langue et au pays (ou à la zone géographique) :
 - Les formats de données :
 - Les dates : noms complets et abrégés des jours de la semaine et des mois de l'année ainsi que la structure de la date (dont le séparateur de date)
 - Les nombres : symboles utilisés comme séparateur de milliers et séparateur décimal et emplacement de ces symboles dans les nombres
 - Les heures : indicateurs utilisés pour les formats horaires en tranches de 12 heures (indicateurs AM et PM par exemple) et structure de l'heure

- Les valeurs monétaires : symboles numériques et monétaires ainsi que l'emplacement de ces symboles dans les valeurs monétaires
- L'ordre de tri : mode de tri des données propre au jeu de codes de caractères et à la langue
- La gestion des chaînes : comprend des tâches comme la comparaison de la casse des lettres (majuscules et minuscules), les sous-chaînes et la concaténation
- Le codage de caractères : mappage d'un caractère (une lettre de l'alphabet) à une valeur numérique d'un jeu de codes de caractères. Par exemple, dans le jeu de codes ASCII la lettre A correspond à 65 tandis que dans le jeu de codes EBCDIC cette lettre est mappée à la valeur 43. Un *jeu de codes de caractères* contient un codage pour tous les caractères d'un ou plusieurs alphabets.

Le nom de l'environnement local respecte le format suivant :

11_TT.codeset

où 11 est un code de langue à deux caractères (habituellement en minuscules, TT est un code de pays à deux caractères (habituellement en majuscules) et codeset désigne le nom du jeu du codes de caractères associé. La partie codeset du nom est facultative. L'environnement local est habituellement installé en même temps que le système d'exploitation.

Installation d'un environnement local

Régler l'environnement local de l'utilisateur par défaut sur un environnement bidirectionnel (arabe ou hébreu) est une condition préalable obligatoire pour permettre le traitement adéquat des données dépendantes de l'environnement local. A défaut, les caractères bidirectionnels ne s'affichent pas correctement (voir *System Installation Guide for Windows* pour plus d'informations). L'environnement local Windows est habituellement iw_IL pour l'hébreu et ar_AE pour l'arabe. Le jeu de caractères ASCII est 1256 pour l'arabe et 1255 pour l'hébreu (voir *System Installation Guide for Windows* pour plus d'informations sur la définition des environnements locaux).

Traitement des données dépendantes de l'environnement local

L'environnement d'exécution Java de la machine virtuelle Java (JVM) représente les données dans le jeu de codes de caractères Unicode. Unicode contient un codage pour les caractères des principaux jeux de codes de caractères (mono et multi-octets). La plupart des composants des produits WebSphere Business Integration sont écrits en Java. Il en découle que quand des données sont transférées entre les composants des produits WebSphere Business Integration, il n'est pas nécessaire de convertir les caractères.

Si des données sont transférées entre une application externe et l'environnement WebSphere Business Integration, elles peuvent l'être dans un jeu de codes mono-octet au lieu d'un jeu de codes double-octet comme Unicode. Des mesures doivent être prises alors pour convertir les caractères d'un format vers l'autre. Pour résoudre ce point, les connecteurs ont été internationalisés pour prendre en charge à la fois les jeux de codes de caractères mono-octet et double-octet afin d'afficher les textes dans une langue donnée. Quand un connecteur envoie des données depuis un lieu qui utilise un jeu de codes de caractères donné vers un courtier, il convertit les données depuis un jeu de codes mono-octet vers un jeu de codes de caractères double-octet (Unicode) car le courtier d'intégration utilise Unicode.

Dans la mesure où les messages d'information et les messages de journal utilisent un jeu de codes mono-octet, le paramètre d'environnement local approprié doit être défini pour que le texte bidirectionnel soit converti correctement entre Unicode et les jeux de caractères mono-octet. Pour que les messages d'information et les messages d'erreur soient enregistrés dans la langue appropriée à votre zone géographique, configurez votre environnement local avec la langue bidirectionnelle appropriée. Pour plus d'informations sur les propriétés de configuration, voir "Establishing a locale" dans le document *System Installation Guide for Windows*.

Remarque : Vous devez choisir la page de codes par défaut appropriée à l'invite du DOS. A défaut, les données affichées ne pourront pas être lues par les éditeurs ou les afficheurs standard Windows (voir "Changing your DOS prompt code page for ASCII settings for Hebrew and Arabic" dans le document *System Installation Guide for Windows*).

Remarques relatives à la conception

Le codage est traité dans l'environnement WebSphere Business Integration à trois niveaux :

- données de contenu ;
- métadonnées et données de configuration ;
- données de fichier journal et de trace.

Le niveau des données de contenu traite les données des instances d'exécution des objets métier. Le niveau des métadonnées et des données de configuration traite les données utilisées par les connecteurs pour établir et gérer les communications entre les applications externes et l'environnement WebSphere Business Integration. Le niveau des données de fichier journal et de trace traite les messages d'information journalisés dans les différents fichiers de trace et journaux. Certains adaptateurs sont plus flexibles que d'autres pour la prise en charge du codage. Par exemple, un connecteur XML peut traiter la spécification de codage contenue dans les en-têtes d'un fichier XML qui permet une prise en charge complète du codage. En revanche, un connecteur pour SAP ne permet qu'une prise en charge limitée du codage fournie par l'API client SAP. Pour le niveau des données de contenu comme pour le niveau des métadonnées et données de configuration, si le connecteur a une option de configuration, le paramètre d'environnement local est remplacé par l'option de spécification de codage. Pour le niveau des données de fichier journal et de trace, le jeu de code par défaut de l'invite DOS est utilisé pour coder les données affichées.

Codage des données de contenu

Il n'existe pas d'approche unique pour le codage des données de contenu entre les connecteurs. Il existe trois approches de base pour traiter la manière dont les connecteurs assurent la prise en charge du codage des données de contenu. Ces approches sont les suivantes :

- Gestion automatique du codage par des API middleware ou tiers utilisées pour la sérialisation et la désérialisation des données des objets métier (par exemple une API client SAP/PS et un pilote JDBC pour SGBDR).
- Utilisation de la technologie sous-jacente qui possède la configuration appropriée pour le connecteur ou le gestionnaire de données (par exemple à l'aide du paramètre charset contenu dans l'en-tête XML pour la spécification XML DH ou MIME par attachement dans le connecteur Email).

- Activation de la gestion interne du codage dans les produits WebSphere Business Integration et spécification via un attribut comme `DataEncoding` du méta-objet métier `MO_JTextConnector_Default` utilisé dans le connecteur JText.

Ces trois approches permettent de gérer les données de contenu avec les adaptateurs (voir le guide de votre adaptateur pour connaître l'approche la plus adaptée).

Codage des métadonnées de configuration

La prise en charge du codage des métadonnées de configuration dépend souvent des capacités du code de l'adaptateur middleware utilisé pour établir une connexion avec une application externe.

Les métadonnées et les données de configuration codées en ASCII dans les outils WebSphere (dans le modèle d'adaptateur ou les modèles d'objet métier pris en charge) sont stockées en Unicode dans le référentiel. Au moment de l'exécution, ces données sont utilisées comme un argument pour les API JAVA natives ou les API tiers afin d'établir une liaison de communication avec une source externe. Le code middleware utilisé pour établir une liaison dans le code de l'adaptateur recouvre des API Java natives comme des modules (par exemple, les connecteurs Email ou WebSphere MQ), ou encore des API tiers (par exemple PeopleSoft). De plus, plusieurs niveaux de code middleware peuvent intervenir entre le code Java de l'adaptateur et la source externe, comme dans le cas de JText et du dossier File System sur les systèmes non Windows. Si un processus Windows accède au dossier File System, l'appel émis par l'API Java peut être traité indirectement par le code qui gère le montage des partitions non Windows au lieu du dossier Windows File System. Dans la mesure où il n'est pas certain que les API Java natives, les API tiers ou les codes middleware prendront en charge une spécification de codage explicite, les produits WebSphere Business Integration ne prennent en charge que le codage Unicode pour les métadonnées et les données de configuration bidirectionnelles.

Codage des données de fichier journal et de trace

Par défaut, le codage des données bidirectionnelles enregistrées dans les fichiers journaux et les fichiers de trace utilise la page de codes de l'invite du DOS qui appelle le processus. Malheureusement, la page de codes par défaut de l'invite DOS associée aux paramètres nationaux pour l'arabe et l'hébreu n'est pas la page de codes standard utilisée pour ces langues sur le système d'exploitation Windows (1256 et 1255). Donc, si la page de codes de l'invite DOS associée aux processus Java de WebSphere Business Integration n'est pas changée, les données bidirectionnelles envoyées par ces processus aux fichiers journaux et aux fichiers de trace ne s'affichent pas correctement dans la plupart des afficheurs Windows qui utilisent les pages de codes standard pour afficher les données en langue bidirectionnelle (voir "Changing your DOS prompt code page" dans le document *System Installation Guide for Windows* pour plus d'informations).

Prise en charge des données bidirectionnelles

Les produits WebSphere Business Integration prennent en charge les données des langues bidirectionnelles. Un script bidirectionnel contient à la fois du texte écrit de droite à gauche et des nombres ou des segments de texte écrits sur le modèle occidental de gauche à droite (par exemple des scripts de type Latin en anglais, français, ou utilisant l'alphabet cyrillique ou grec).

L'arabe et l'hébreu sont les deux principaux groupes de langues qui utilisent des scripts bidirectionnels. Le groupe Arabe comprend l'arabe, le farsi (perse), l'urdu et d'autres langues. Le groupe Hébreu comprend l'hébreu mais aussi le Yiddish et le Ladino. Les deux groupes de langues possèdent un alphabet (27 lettres uniquement) et peuvent utiliser un système de codage mono-octet.

Caractéristiques des langues bidirectionnelles

Deux caractéristiques distinguent les scripts bidirectionnels des scripts en langue occidentale (anglais, français, allemand, grec, etc.). Ces caractéristiques sont la *bidirectionnalité* et la *forme*.

Bidirectionnalité

La bidirectionnalité fait appel à sept concepts clés :

- la segmentation ;
- l'imbrication ;
- l'orientation globale ;
- l'ordre logique/physique du texte ;
- les typages de texte et les méthodes de réorganisation associées ;
- la permutation symétrique ;
- la mise en miroir des objets des interfaces traduites.

Remarque : Dans tous les exemples ci-dessous, les lettres en capitales comme DCBA représentent des lettres de l'alphabet arabe ou hébreu.

Segmentation

Un segment se définit comme une chaîne contenant une partie de texte à l'intérieur d'une autre chaîne ayant une directionnalité donnée. La partie principale d'un script peut avoir une orientation de droite à gauche tandis qu'une autre partie peut être orientée de gauche à droite. Prenons comme exemple de segmentation bidirectionnelle une adresse postale comme Entrance B 25 Maple Street. Ecrite en hébreu, cette adresse devient B ECNARTNE 25 TEERTS ELPAM. Dans cet exemple, les parties B ECNARTNE et TEERTS ELPAM de la chaîne de texte sont orientées de droite à gauche mais le nombre 25 est orienté de gauche à droite.

Imbrication

Un segment imbriqué est un segment de texte qui possède une directionnalité donnée mais contient aussi un autre segment dont la directionnalité est opposée. Reprenons l'exemple de l'adresse B ECNARTNE 25 TEERTS ELPAM. Cette adresse possède un niveau d'imbrication. Le nom de la rue, TEERTS ELPAM, se lit de droite à gauche mais le flux s'inverse ensuite pour l'entrée du numéro 25 qui se lit de gauche à droite. Après le numéro, l'orientation de la chaîne redevient de droite à gauche pour B ECNARTNE.

Orientation globale

L'orientation globale, qu'on appelle aussi sens d'écriture, sens de lecture, ou direction du paragraphe, désigne le côté de l'écran, de la fenêtre ou de la page où commence le texte. L'orientation globale du texte est aussi *contextuelle*. Cela veut dire que la signification du texte dépend de son contexte. Pour illustrer cette dépendance contextuelle, prenons un texte bidirectionnel. Lisez la phrase qui suit :

FRED DOES NOT BELIEVE taht yas syawla i.

Cette phrase a un sens si on la lit de gauche à droite (Fred does not believe I always say that / Fred ne croit pas que je dis toujours ça) et un autre quand on la lit de droite à gauche (I always say that Fred does not believe / Je dis toujours que Fred ne croit pas). Comme l'orientation globale n'est pas toujours évidente selon le contexte, le développeur d'applications doit savoir comment le texte sera lu (de gauche à droite ou l'inverse).

Ordre physique et logique du texte

Un texte bidirectionnel peut être ordonné sur le mode logique ou physique. Dans les environnements de poste de travail, le mode de saisie habituel du texte bidirectionnel est l'ordre logique car le texte est traité comme le texte Latin. Si vous utilisez l'ordre logique, vous devez fournir un moyen d'inverser les segments dont la direction est opposée à l'orientation globale. Par exemple, si l'orientation globale est celle de l'anglais (de gauche à droite), les segments en arabe et en hébreu demandent un traitement spécial dans leur direction native, de droite à gauche. Inversement, pour saisir et traiter le texte bidirectionnel dans les grands systèmes informatiques, on utilise habituellement l'ordre physique. Il en découle que quand vous intégrez du texte bidirectionnel depuis des environnements d'ordinateur principal ou de poste de travail, vous devez le transformer de manière à ce que tout le texte ait le même ordre.

L'ordre logique et physique du texte utilisé dans les scripts bidirectionnels est également important. L'ordre physique se rapporte à la manière dont le segment de texte se présente physiquement tandis que l'ordre logique concerne la manière dont il est saisi (ou lu). Selon la situation, certains segments doivent être réorganisés en ordre logique ou physique. Par exemple, prenons la phrase suivante :

my wife's name is ILIN (le nom de ma femme est ILIN)

L'orientation globale de cette phrase est de gauche à droite. On lit ce texte en commençant par la première lettre m, suivie de y, et ainsi de suite. Dans l'ordre physique, les lettres i et s sont suivies par la lettre I du segment ILIN, mais en hébreu ILIN se prononce NILI, par conséquent dans l'ordre logique la première lettre de ce segment est N et pas I.

Typage de texte

Le typage de texte est l'approche la plus appropriée pour enregistrer un texte particulier. Cela signifie que les différents types de texte demandent des techniques d'enregistrement spécifiques. Il existe trois types de texte pour l'enregistrement : visuel, implicite (logique) et explicite.

Le type de texte visuel est la plus ancienne des méthodes d'enregistrement, il s'agit d'une simple copie du contenu de l'écran de saisie. Cette forme de typage de texte demande au programmeur de savoir où se situent les segments imbriqués et comment les traiter. De nombreuses applications et leurs fichiers utilisent ce type de texte.

L'enregistrement dans le type de texte implicite suppose que les lettres de l'alphabet latin ont une directionnalité de gauche à droite et que les alphabets arabe, farsi, urdu et hébreu ont une directionnalité de droite à gauche. Pour gérer la bidirectionnalité, on utilise un algorithme de traitement du texte implicite afin de reconnaître les segments sur la base de leurs caractéristiques directionnelles et permettre l'inversion automatique des segments. La principale limite du type de texte implicite est son incapacité à traiter les chaînes dans lesquelles sont mélangés des chiffres et des lettres (de gauche à droite et dans l'autre sens) comme dans le cas des numéros de pièce par exemple.

L'enregistrement avec le type de texte explicite suppose que des caractères de commande supplémentaires sont imbriqués dans une chaîne de texte pour demander à l'algorithme d'exécuter des inversions des segments, des mises en forme ou des sélections numériques et d'autres transformations. La limitation du type de texte explicite est nécessaire pour que les processus automatiques puissent gérer les commandes imbriquées. Il existe une technique spéciale pour lier les types de texte implicite et explicite. Cette technique est l'*algorithme d'affichage de base* qui est défini dans l'algorithme Unicode Standard Bidi.

Permutation symétrique

La permutation symétrique est la capacité de gérer les caractères comme <, (, [, { , qui possèdent un caractère symétrique complémentaire ayant la direction opposée, par exemple >,),], }. Ces caractères sont problématiques car une permutation globale transformerait par exemple A > B en B > A. La permutation symétrique permet la conversion du symbole uniquement pour obtenir B < A.

Mise en miroir des objets d'interface

La mise en miroir des objets d'une interface utilisateur graphique traduite transforme l'interface en fonction de la directionnalité de la langue. Par exemple, la mise en miroir des objets peut déplacer les boutons de menu et l'arborescence de navigation de la gauche vers la droite. Dans le cas contraire, les cadres et les fenêtres ne seraient pas mis en miroir. La figure 44 illustre la mise en miroir des objets d'un menu déroulant.

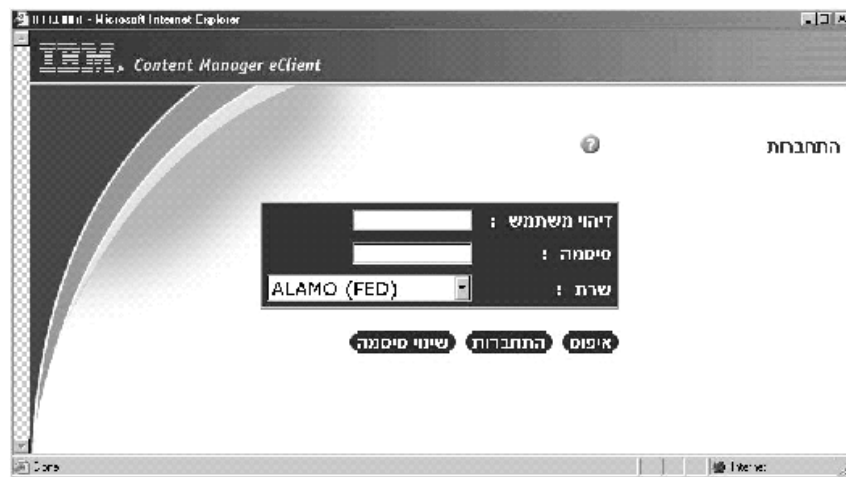


Figure 44. Fenêtre mise en miroir contenant des libellés bidirectionnels

Mise en forme

La forme des caractères est une caractéristique de nombreuses langues complexes, en particulier des langues **cursives** que sont l'arabe et l'hébreu. Un système d'écriture est dit "cursif" s'il contient des caractères contigus et reliés les uns aux autres dans un mot, et s'il est conçu pour l'écriture manuelle plus que pour l'impression. En arabe par exemple, certaines lettres ne peuvent s'attacher qu'à une lettre située à droite. De plus, certaines lettres peuvent changer de forme selon leur position dans le mot et les propriétés de rattachement des lettres contiguës. Ces aspects font que la mise en forme est importante pour afficher correctement un texte bidirectionnel. Le processus de mise en forme affiche les caractères sous la forme qui convient en remplaçant une représentation abstraite d'un caractère par la

forme appropriée. Pour cela, on utilise la forme de base d'un caractère afin de permettre la sélection d'un caractère cursif donné sans spécifier sa forme.

Une routine de détermination de forme permet ensuite au logiciel ou à l'utilisateur de choisir automatiquement (par un algorithme) la forme appropriée du caractère en fonction du contexte. Dans la plupart des cas, les formes de base d'un texte en langue cursive sont enregistrées. La mise en forme a deux autres caractéristiques. Il s'agit de la composition des caractères et des nombres nationaux.

La composition des caractères se définit comme la correspondance entre le nombre de caractères stockés et le nombre de caractères affichés. Pour garder cette correspondance, on utilise des **ligatures** et des **signes diacritiques**. On utilise des ligatures quand deux caractères ou plus peuvent être représentés par un caractère unique qui n'occupe qu'une seule cellule. Les signes diacritiques présents dans les scripts bidirectionnels sont des marques associées à une consonne dans une orientation donnée (au dessus, dans, sous ou à côté) afin de représenter des voyelles. Quand ces marques sont stockées, elles occupent un emplacement physique mais si elles sont utilisées pour la représentation, elles peuvent occuper la même cellule que les consonnes associées. En langue arabe, des signes diacritiques d'espacement sont utilisés comme caractères séparés qui sont affichés en fonction du caractère auquel ils se rapportent.

Les nombres nationaux demandent également un traitement spécial car ils sont utilisés différemment selon les langues. Par exemple, en hébreu les nombres sont représentés par des chiffres arabes (1,2,3...0). En revanche, les langues cursives comme l'arabe, le farsi et l'urdu ont leur propres glyphes pour représenter les chiffres. Le libellé des chiffres dans les langues cursives utilise soit les chiffres indiens soit les chiffres arabo-indiens. Qu'il s'agisse du système de chiffres arabe, indien ou arabo-indien, tous les nombres simples sont représentés de gauche à droite mais le sens des formules mathématiques peut varier d'une langue à l'autre. Par exemple, en arabe, les formules mathématiques s'écrivent de gauche à droite tandis qu'elles s'écrivent de droite à gauche en farsi. Donc, si les nombres sont habituellement stockés sous forme de chiffres arabes, ils peuvent être représentés par des glyphes nationaux ou des chiffres arabes selon l'intention de l'utilisateur ou du développeur.

Attributs et transformations de la présentation

L'utilisation de scripts bidirectionnels demande une attention particulière sur les points suivants :

- «Présentation du texte»
- «Attributs de présentation» à la page 111
- «Transformations de présentation» à la page 112

Présentation du texte

Le texte bidirectionnel peut avoir différentes présentations. Une présentation change en fonction des segments contenus dans le texte et, comme dans le cas des scripts en arabe, elle change aussi dans la forme des caractères et des nombres utilisée. Les transformations entre les différentes présentations demandent des fonctions de transformation également appelées fonctions de présentation ou fonctions de services de présentation.

Attributs de présentation

Pour définir les caractéristiques de la présentation d'un texte bidirectionnel, il faut une série d'attributs. Les attributs bidirectionnels permettent de garantir la présentation du texte et son mode de transformation. Ces attributs sont généralement externes au texte et sont stockés dans un fichier de ressources externe. Les cinq attributs bidirectionnels sont les suivants :

- Orientation
- Type de texte
- Mise en forme du texte
- Permutation symétrique
- Formes des nombres

Orientation

L'orientation désigne la limite de la zone de présentation (fenêtre, cadre, page) dans laquelle commence l'écriture du texte bidirectionnel. Ceci repose sur la directionnalité du premier caractère du texte. La directionnalité peut être de droite à gauche, de gauche à droite ou contextuelle.

Type de texte

Le type de texte détermine le genre d'algorithme utilisé lors de la transformation de la présentation du texte. Il existe trois types d'algorithme : visuel, implicite, explicite. Un algorithme visuel copie des lignes entières de texte comme elles apparaissent indépendamment des segments directionnels imbriqués pouvant exister. Un algorithme implicite reconnaît les segments directionnels sur la base de la directionnalité naturelle des caractères (par exemple, de droite à gauche pour l'arabe, de gauche à droite pour l'anglais) et prépare les inversions de segment en conséquence. Un algorithme explicite reconnaît les segments directionnels et exécute les inversions sur la base des commandes visuelles, explicites et directionnelles imbriquées dans le texte.

Mise en forme du texte

La mise en forme du texte est importante pour les scripts en arabe car les caractères y prennent différentes formes en fonction de leur position dans un mot tout comme les marques de liaison associées aux caractères contigus.

Mise en forme symétrique

La mise en forme symétrique (également appelée permutation symétrique) détermine quand des caractères spécifiques comme (, >, [, { doivent être inversés), <,], } pour préserver le sens logique du texte.

Formes des nombres

La forme des nombres détermine si les nombres présents dans un script en arabe doivent prendre la forme des chiffres européens ou des chiffres arabo-indiens.

Il n'existe pas de combinaison de valeurs possibles prédominante pour les attributs de présentation du texte bidirectionnel. Les applications existantes traitent les données avec différentes combinaisons possibles de ces valeurs. Quand une chaîne de données bidirectionnelle est transmise à une application, il est important que cette application puisse reconnaître les attributs de texte associés.

Transformations de présentation

Le texte bidirectionnel est stocké et traité dans différents environnements (plates-formes) et dans différentes présentations. Pour transformer une présentation, vous avez besoin de fonctions de transformation de présentation. Les produits WebSphere Business Integration utilisent des fonctions de transformation de présentation basées sur l'algorithme Unicode BiDi. Vous trouverez cet algorithme à l'adresse suivante :

<http://www.unicode.org/reports/tr9/>)

ainsi qu'à l'adresse suivante (dans IBM Java SDK 1.4.1) :

<http://www-106.ibm.com/developmentworks/java/jdk/bidirectional/JAVABIDI.htm>

Prise en charge des scripts bidirectionnels dans les produits WebSphere Business Integration

La prise en charge des scripts bidirectionnels dans les produits WebSphere Business Integration intervient à différents niveaux et au travers des configurations des différents composants. La prise en charge des langues bidirectionnelles est actualisée sur les trois niveaux.

- Affichage, typage, enregistrement et récupération des caractères des scripts bidirectionnels créés avec les outils WebSphere (par exemple, Connector Configurator, BO Designer, System Manager, etc).
- Utilisation de la conversion de page de codes pour convertir les caractères bidirectionnels d'un format dans un autre (entre le jeu de codes Unicode et un jeu de codes mono-caractère).
- Utilisation des transformations de texte bidirectionnel pour la conversion entre le format bidirectionnel Windows (utilisé dans l'environnement WebSphere Business Integration) et les formats de texte bidirectionnels utilisés dans les applications externes.

Les produits WebSphere Business Integration suivants peuvent prendre en charge le traitement des données des scripts bidirectionnels : Toolset, Adapter Framework, ICS Broker, ainsi que neuf adaptateurs (JText, JDBC, Email, XML, WebSphere MQ, SAP, PeopleSoft, Web services et Lotus Domino) et les ODA associés (JDBC, XML, WSDL).

Parmi les composants ci-dessus, ceux du premier niveau font partie du WebSphere Business Integration Toolset, des outils Java tels que System manager, des outils C++ comme Business Object Designer et Connector Configurator, et des outils basés sur le Web comme Dashboard. Les composants du deuxième niveau sont les adaptateurs prenant en charge le texte bidirectionnel. Les composants du troisième niveau sont les adaptateurs prenant en charge le texte bidirectionnel et Adapter Framework.

Les connecteurs

Les connecteurs opèrent une médiation entre une application et une ou plusieurs collaborations. Ils traitent les données de contenu bidirectionnel issues d'une application et utilisent les métadonnées et les données de configuration définies par les outils WebSphere comme Connector Configurator, pour établir une communication entre une application qui utilise le format bidirectionnel natif et une ou plusieurs collaborations.

Adapter Framework

Le produit Adapter Framework procure une structure qui relie une application à une ou plusieurs collaborations et gère le transfert des

données de contenu entre une application et une collaboration. Adapter Framework prend en charge les langues bidirectionnelles et assure la cohérence des données de contenu bidirectionnel.

Toolset

La boîte à outils de WebSphere Business Integration fournit des outils d'administration et de développement utilisables avec InterChange Server. Les outils de la boîte à outils peuvent prendre en charge de manière implicite le traitement des données des scripts bidirectionnels et ne demandent pas de configuration supplémentaire hormis pour l'environnement local bidirectionnel (voir *System Installation Guide for Windows* pour plus d'informations).

La section qui suit donne une présentation générale de la prise en charge des données bidirectionnelles pour les composants suivants :

- «Prise en charge des scripts bidirectionnels»
- «Prise en charge des scripts bidirectionnels par Adapter Framework»
- «Prise en charge des collaborations pour les scripts bidirectionnels» à la page 114
- «Prise en charge des scripts bidirectionnels par les mappes» à la page 115

Prise en charge des scripts bidirectionnels

Un **connecteur** sert d'intermédiaire entre une application (ou toute autre entité par programmation, tel un serveur Web) et une ou plusieurs collaborations. Un connecteur peut être propre à une application (telle que SAP R/3 version 4) ou à une technologie, telle qu'un format de données ou un protocole (XML ou EDI). Les communications entre les connecteurs et les collaborations peuvent prendre deux formes :

- les **notifications d'événement** des applications qui sont transmises aux connecteurs par certaines applications ;
- le **traitement des requêtes** que les connecteurs exécutent au bénéfice des collaborations.

(Voir Chapitre 5, «Connecteurs», à la page 67 pour plus d'informations).

Les connecteurs traitent les données de contenu issues d'une source de données (application ou programmation) à l'aide de métadonnées et de données de configuration définies avec un des outils WebSphere tel que Connector Configurator. Dans l'environnement des produits WebSphere, les métadonnées et les données de configuration sont représentées et stockées dans le format standard Windows des données bidirectionnelles. Dans la mesure où une application externe peut contenir des métadonnées et des données de configuration dans un format bidirectionnel autre que le format bidirectionnel Windows utilisé dans les produits WebSphere Business Integration, une transformation est requise pour établir une communication entre l'environnement WebSphere Business Integration et l'application externe. Les connecteurs qui prennent en charge les langues bidirectionnelles peuvent être configurés par la propriété de connecteur standard BiDi.Metadata pour appliquer le format bidirectionnel applicable aux métadonnées et aux données de configuration de l'application externe.

Prise en charge des scripts bidirectionnels par Adapter Framework

Adapter Framework relie un connecteur à InterChange Server et les adaptateurs WebSphere Business Integration pour que les contenus puissent circuler entre une application ou un programme et une ou plusieurs collaborations. Quand les

connecteurs traitent les métadonnées et les données de configuration, Adapter Framework traite le contenu des données proprement dit, les valeurs des attributs des objets par exemple. Le composant Adapter Runtime est commun à tous les connecteurs.

Le rôle d'Adapter Framework est alors de préserver la cohérence des données entre la présentation de l'environnement WebSphere Business Integration, dans lequel les données sont au format bidirectionnel Windows, et les représentations des applications externes pouvant utiliser d'autres formats bidirectionnels. Si des données sont envoyées de l'environnement WebSphere Business Integration vers une application externe, Adapter Framework effectue les transformations requises, le cas échéant, depuis le format bidirectionnel Windows vers un format bidirectionnel externe. Inversement, si le format bidirectionnel externe n'est pas le format bidirectionnel Windows utilisé dans l'environnement WebSphere Business Integration, Adapter Framework exécute les transformations requises vers le format bidirectionnel Windows.

Adapter Framework peut aussi préserver la cohérence des données quand un courtier autre que le courtier WebSphere Business Integration est utilisé. Si le courtier utilise un format bidirectionnel autre que celui de Windows, une option permet de remplacer ce format par défaut par celui qu'utilise le courtier.

Prise en charge des collaborations pour les scripts bidirectionnels

Dans l'implémentation des produits Websphere, le terme **collaborations** désigne des modules logiciels qui contiennent du code et une logique de processus métier qui gère les interactions entre les applications. Une collaboration peut être simple, et ne comporter que quelques étapes, ou complexe, et comprendre plusieurs étapes et d'autres collaborations.

Les collaborations peuvent être réparties entre un nombre illimité d'applications, et prendre en charge des interventions sur appels synchrones et asynchrones, ainsi que des processus métier de longue durée.

La prise en charge des scripts bidirectionnels par les produits WebSphere couvre les collaborations. Les collaborations réceptionnent des données depuis l'environnement WebSphere (connecteurs, interface d'accès ou autres collaborations) ou depuis d'autres sources externes comme les services Web. Le format de données bidirectionnel est mis en oeuvre implicitement, par l'un des connecteurs pouvant le faire, ou explicitement, par la prise en charge des langues bidirectionnelles ou une API. Si des données arrivent d'un composant qui ne prend pas en charge les langues bidirectionnelles, comme un service Web ou un connecteur, des incohérences de format peuvent exister et entraîner l'échec de la logique des collaborations ou des résultats incorrects. Vous pouvez éviter ce type d'erreur de la manière suivante :

- en acceptant les entrées venant des sources prenant en charge les langues bidirectionnelles ou qui ont le même format que les produits WebSphere Business Integration ;
- en activant la prise en charge des langues bidirectionnelles dans les connecteurs compatibles afin de mettre en oeuvre le format bidirectionnel des données transmises aux collaborations ;
- en utilisant les API BiDi de la classe CwBidiEngine pour mettre en oeuvre le format bidirectionnel des données utilisées ou entrées dans le domaine des

produits WebSphere par une source de données externe (voir le chapitre sur CwBidiEngine dans le document *Collaboration Development Guide*).

Prise en charge des scripts bidirectionnels par les mappes

Les produits WebSphere Business Integration prennent en charge les mappes contenant des scripts bidirectionnels. Les mappes reçoivent des données des connecteurs ou de sources de données externes. Les données qui arrivent dans l'environnement WebSphere Business Integration depuis un connecteur prenant en charge les langues bidirectionnelles ont ainsi la garantie d'avoir un format de langue bidirectionnelle standard (Windows). Toutefois, des données peuvent être introduites dans une mappe depuis des sources externes inconnues, par exemple, des données exportées via un service Web. Dans l'hypothèse où un service Web utilise des données bidirectionnelles qui n'ont pas le format Windows, deux résultats sont possibles. Le premier est que la connexion à ce service risque d'échouer. Le deuxième résultat est que les données bidirectionnelles dont le format n'est pas celui de Windows donnent des résultats inattendus au cours du traitement car elles seront comparées aux données au format bidirectionnel Windows (voir *Using bidirectional functionality in Activity Editor* dans le document *Map Development Guide*). Pour éviter ces erreurs, voir «Prise en charge des collaborations pour les scripts bidirectionnels» à la page 114.

Gestion du texte bidirectionnel

Dans certaines situations, une attention particulière doit être portée au traitement du texte bidirectionnel. C'est le cas, par exemple, quand vous migrez des données de référentiel depuis des versions antérieures de WebSphere Business Integration qui ne prennent pas en charge les langues bidirectionnelles (voir «Migration de données» pour plus d'informations). Sans mesure adéquate, des données dans deux formats de données bidirectionnelles différents peuvent cohabiter dans le même référentiel et entraver le bon fonctionnement de la logique de traitement. C'est aussi le cas quand vous transformez un texte bidirectionnel contenant des modèles spéciaux comme des adresses URL de sites FTP et des adresses de messagerie (voir «API BiDi» à la page 116 pour plus d'informations).

Migration de données

Vous devez prendre certaines précautions quand vous migrez des données depuis des versions plus anciennes des produits WebSphere Business Integration.

Au cours du processus de migration, les données bidirectionnelles de l'ancienne version peuvent persister et être utilisées en même temps que les nouvelles données bidirectionnelles introduites par les connecteurs qui prennent en charge les langues bidirectionnelles. Si cette situation se produit, il n'est pas certain que les données bidirectionnelles manipulées au niveau du serveur seront au format Windows. Les données bidirectionnelles, par exemple des collaborations ou des mappes, peuvent s'en trouver irrémédiablement altéré après le traitement.

Avant d'effectuer la migration des données vers la dernière version des produits WebSphere, il est conseillé de convertir toutes les données présentes dans le référentiel au format bidirectionnel Windows au moyen des API BiDi de cette version. (Voir le chapitre CxBidiEngine dans le document *Map Development Guide* pour plus d'informations).

Chaînes bidirectionnelles spéciales

Les adresses FTP et e-mail sont des exemples de situations dans lesquelles l'application explicite de transformations bidirectionnelles peut entraîner une interprétation incorrecte des données. Pour que l'interprétation soit correcte, ces chaînes sont analysées avant la transformation et leurs sous-composants problématiques sont identifiés. La chaîne est ensuite décomposée et une transformation bidirectionnelle est appliquée à chaque sous-composant concerné. Une fois la transformation terminée, les sous-composants sont réunis en une chaîne unique qui représente la valeur transformée correcte. Cette valeur est ensuite enregistrée pour utilisation ultérieure. On utilise ce processus notamment pour le traitement des méta-objets métier (MTO-Meta Business Object).

API BiDi

Les produits WebSphere Business Integration sont fournis avec des API de la classe BiDi qui permettent de mettre en oeuvre un format bidirectionnel pour les différents composants. Les fonctions des API BiDi peuvent être employées dans Adapter Framework pour mettre en oeuvre le format bidirectionnel des données de contenu, dans les connecteurs pour appliquer celui des métadonnées et des données de configuration, et dans les collaborations et les mappes pour convertir au format bidirectionnel les données importées d'une source externe.

Méthodes des API BiDi

Les API BiDi proposent trois méthodes :

- BiDiBusObjTransformation
- BiDiBOTransformation
- BiDiStringTransformation

Fonction BiDiBusObjTransformation

La méthode BiDiBusObjTransformation transforme les objets de type BusinessObject d'un format bidirectionnel dans un autre. Cette méthode s'utilise pour les collaborations (voir le chapitre CxBiDiEngine dans le document *Collaboration Development Guide* pour plus d'informations).

Fonction BiDiBOTransformation

La fonction BiDiBOTransformation s'applique aux instances des objets de type BusinessObject. Elle s'utilise avec Adapter Framework (voir le chapitre CxBiDiEngine dans le document *Collaboration Development Guide* ou *Map Development Guide* pour plus d'informations).

Fonction BiDiStringTransformation

La fonction BiDiStringTransformation s'applique aux objets de type String. Elle s'utilise avec les objets internes et externes à l'environnement des produits WebSphere (voir le chapitre CxBiDiEngine dans le document *Collaboration Development Guide* ou *Map Development Guide* pour plus d'informations).

limitations conceptuelles

La conception actuelle n'offre qu'une solution limitée pour la prise en charge des langues bidirectionnelles. Voici quelques unes de ces limitations :

- La prise en charge des données bidirectionnelles pour les métadonnées se fait via un paramètre unique applicable à toutes leurs propriétés bidirectionnelles. Cela signifie que quand un attribut repose sur un paramètre bidirectionnel unique, soit tous les attributs de métadonnées sont transformés soit aucun ne l'est. Vous ne pouvez pas définir des formats bidirectionnels différents pour

chacune des propriétés bidirectionnelles de métadonnées ou spécifier quels paramètres de métadonnées doivent être exemptés de ces propriétés.

- Dans la mesure où tous les composants des produits WebSphere Business Integration ne prennent pas en charge les langues bidirectionnelles, par exemple Server Access Interface et les différents connecteurs, dans le cas où ces composants sont utilisés avec d'autres qui les prennent en charge, il n'est pas garanti que les données d'InterChange Server ou des adaptateurs WebSphere Business Integration auront le même format de données bidirectionnelles. Cette situation peut induire une représentation incohérente des données bidirectionnelles dans InterChange Server ou dans les adaptateurs WebSphere Business Integration et un traitement incorrect de ces données. Il n'existe pas de mécanisme pour assurer l'uniformité du format des données bidirectionnelles dans InterChange Server ou dans les adaptateurs WebSphere Business Integration. Il revient à l'utilisateur de mettre en oeuvre ce format. Pour appliquer le format bidirectionnel aux données reçues des composants qui ne prennent pas en charge les langues bidirectionnelles, l'utilisateur peut employer les API BiDi.
- La prise en charge du contenu bidirectionnel se fait au niveau du connecteur. Vous ne pouvez utiliser qu'une seule spécification de format bidirectionnel par connecteur. En d'autres mots, tous les objets métier pris en charge par un connecteur donné sont supposés contenir des données bidirectionnelles dans le même format.
- Les produits WebSphere Business Integration prennent en charge les langues bidirectionnelles quand la configuration manuelle adaptée a été réalisée (voir *System Installation Guide for Windows* pour plus d'informations).
- Les produits WebSphere Business Integration contiennent neuf adaptateurs qui prennent en charge les langues bidirectionnelles : JText, JDBC, Email, XML, WebSphere MQ, SAP, PeopleSoft, Web services et Lotus Domino.
- Choisir d'aligner le format bidirectionnel des données internes des produits WebSphere Business Integration sur le format bidirectionnel Windows est judicieux car les outils des produits WebSphere comme Connector Configurator, Map Designer, BO Designer, etc, ne fonctionnent que sur la plateforme Windows. Si la prise en charge des outils doit couvrir d'autres plates-formes, la conception actuelle ne sera pas suffisante et devra être modifiée.
- Choisir le format bidirectionnel Windows comme format interne pour InterChange Server et les adaptateurs WebSphere Business Integration limite la communication avec les applications qui utilisent d'autres formats bidirectionnels. Par exemple, deux applications utilisant un format bidirectionnel visuel pour communiquer via les produits WebSphere Business Integration qui possèdent, dans leur conception, un format bidirectionnel logique interne. Notez que la transformation bidirectionnelle n'est pas transitive.
- WebSphere ne prend pas en charge les caractères bidirectionnels dans les noms des objets métiers ou de leurs attributs. Après la création d'un modèle d'objet métier basé sur des fichiers XML/XSD ou de tables de base de données dont un attribut, un élément ou un nom de colonne contient des caractères bidirectionnels, vous devez renommer tous les objets métier et leurs attributs qui contiennent des caractères non latins afin qu'ils ne comprennent que des caractères latins.
- Actuellement, seuls les gestionnaires de données délimitées et XML sont pris en charge. Le gestionnaire de données Request-Response n'est pris en charge que s'il utilise des gestionnaires de données délimitées et XML ou les utilise séparément pour gérer les échanges Request-Response. Les gestionnaires de données EDI, Fixed Width et Name-Value ne sont pas pris en charge.

- WebSphere ne prend pas en charge les caractères non latins, notamment les caractères bidirectionnels, contenus dans les noms des entités de base, par exemple Connector, Collaboration, Map, Business Object, Business Object, etc.

Récapitulatif

Ce chapitre présentait les caractéristiques du texte bidirectionnel et expliquait leur prise en charge dans le système WebSphere Business Integration. Les points clés à retenir sont les suivants :

- Les scripts bidirectionnels contiennent du texte écrit de gauche à droite et des segments imbriqués écrits de droite à gauche.
- Le texte bidirectionnel a deux caractéristiques de base ; la bidirectionnalité et la forme. Chaque caractéristique doit être prise en compte quand on développe une application en langue bidirectionnelle.
- Un environnement local doit être défini pour pouvoir utiliser des scripts en langue bidirectionnelle.
- InterChange Server prend en charge les langues bidirectionnelles si une configuration système appropriée est mise en oeuvre (voir *System Installation Guide for Windows* pour plus d'informations). Les adaptateurs WebSphere Business Integration comptent neuf adaptateurs qui prennent en charge les langues bidirectionnelles.
- Le format bidirectionnel interne des produits WebSphere Business Integration est le format bidirectionnel standard Windows (logique de gauche à droite).

Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non disponibles dans certains pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Cependant, 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 décrits 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

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales :

LE PRÉSENT DOCUMENT EST LIVRÉ EN L'ÉTAT. IBM DÉCLINE TOUTE RESPONSABILITÉ, EXPLICITE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE NON-CONTREFAÇON ET D'APTITUDE À L'EXÉCUTION D'UN TRAVAIL DONNÉ. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

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

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

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

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

IBM Burlingame Laboratory Director
IBM Burlingame Laboratory
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. IBM 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.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples peuvent mentionner 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.

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.

Informations sur l'interface de programmation

Les informations relatives à l'interface de programmation, si elles vous ont été fournies, sont destinées à vous aider à créer des logiciels d'application avec ce programme.

Les interfaces de programmation génériques vous permettent d'écrire les programmes d'application qui utilisent les services des outils de ce programme.

Néanmoins, ces informations peuvent également contenir des informations sur les diagnostics, les modifications et l'optimisation. Les informations sur les diagnostics, les modifications et l'optimisation vous sont fournies pour vous aider lors du débogage de votre logiciel d'application.

Remarques : N'utilisez pas ces informations sur les diagnostics, les modifications et l'optimisation dans le cadre de votre programmation, car elles sont susceptibles d'être modifiées.

Marques de commerce 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 :

IBM
le logo IBM
AIX
CrossWorlds
DB2
DB2 Universal Database
Domino
Lotus
Lotus Notes
MQIntegrator
MQSeries
Tivoli
WebSphere

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux États-Unis et/ou dans certains autres pays.

MMX, Pentium et ProShare sont des marques d'Intel Corporation aux Etats-Unis et/ou dans certains autres pays.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

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

System Manager et Adapter Monitor contiennent des logiciels développés par Eclipse Project (<http://www.eclipse.org/>).



Index

A

Agent du connecteur 3, 17
 construction d'objets métier 74, 78
 création 82
 démarrage 68
 emplacement de 17
 envoi d'un objet métier 75
 interrogation et 73
 traitement des événements 73
 traitement des requêtes 75, 78
 vérification des abonnements 74
API
 texte bidirectionnel 116
API (interface de programme
 d'application) JDBC (Java Database
 Connectivity) 26
API de transformation bidirectionnelle
 fonction
 BiDiBusObjTransformation 116
API de transformation bidirectionnelle,
 fonction BiDiBOTransformation 116
Application
 basée sur des événements 70
Archivage des événements 75
Assistant d'étalonnage 46
Asymétrique 34
Asymétrique et dynamique 34
Attribut (d'objet métier) 21, 60, 62
 composé 61
 informations propres à l'application
 pour 63
 organisation 63
 propriétés 61
 simple 61
 types de données de 61
Attribut (objet métier)
 transféré par mappage 88
Attribut ObjectEventId 61
Authentification 37

B

Base de données
 fournisseurs pris en charge 26
 référentiel 26
 transactions dans 92
Basé sur les rôles 37
Bidirectionnalité
 définie 107
Boîte à outils WebSphere Business
 Integration 3
Business Object Designer 46

C

Caractéristiques
 bidirectionnelles 107
Caractéristiques bidirectionnelles 107
Chiffrement 33
Client d'accès 4, 10

Codage des données
 métadonnées de configuration 106
Collaboration 3, 47
 abonné à des objets métier 10
 configuration 45
 configurer 48
 connecteur et 56, 67
 déclencheur 10
 groupe 50
 initialisation 58
 intervention sur appel 11, 49
 liaison 17, 50
 longue durée 49
 modèle 45, 47, 92
 objet 47
 ports 50
 processus métier 53, 56
 propriétés 47
 reprise sur incident 100
 scénario 53
 serveur d'informations d'objets
 métier 10
 traitement 48
 traitement simultané 49
 transactionnelle 28, 91
Collaboration transactionnelle 28, 91,
 101
 fenêtre de vulnérabilité 98
 invalidation 95
 isolement dans 97, 98
 modèle pour 92
 niveaux de transaction 98, 100
 reprise sur incident 100
Common Object Request Broker
 Architecture (CORBA) 30
Complète 34
Composants d'intégration 45
Conception
 codage des données de contenu 105
Confidentialité 34
Confidentialité complète 34
Configuration
 collaboration 45, 48
 connecteur 81
 Le système InterChange Server 44
Connecteur 3, 16, 67
 basé sur les métadonnées 80
 collaborations et 56, 67
 communication avec une
 application 17
 comportement de la notification
 d'événement 69
 comportement du traitement des
 requêtes 75, 78
 composants 17
 configuration 69, 81
 développement 82
 initialisation 67
 modification 82
 multiprocessus 78
 niveau de transaction 99

Connecteur (*suite*)
 propriétés 46, 82
 serveur d'informations d'objets
 métier 10
 structure à connecteur client 17
Connecteurs
 langues bidirectionnelles 113
 prise en charge des données
 bidirectionnelles 113
Connectivité 3, 6
Connector Configurator 46, 81, 82
contrôle d'accès 37
Contrôle d'accès 37
Contrôle d'accès basé sur les rôles 37
Contrôleur de connecteur 3, 17, 26
 démarrage 67
 emplacement de 17
 en tant que service InterChange
 Server 26
 interaction de l'utilisateur avec 26
CORBA (Common Object Request Broker
 Architecture) 30
Correction 95

D

Déclencheur 17
 appels directs 10
 événements 10
Définition d'un objet métier 59
 association avec un événement 74
 composants 60
 téléchargement par un connecteur de
 la 69
Définition de relation 87, 89
 identité 89
 non-identité 89
 Valeur recherchée 89
Distribution Internet 32
Données dépendantes de l'environnement
 local
 traitement 104
Dynamique 34

E

Environnement de test 45
Environnement local
 installation 104
 prise en charge 103
 remarques relatives à la
 conception 105
Étape d'activité (scénario) 55, 56
Étape de la sous-transaction 93
Étape de sous-transaction 95
 illustration de 94
Événement 19, 70
 archivage 75
 boîte de réception 70
 déclenchement 10

Événement (*suite*)
déttection 73
liaison 17
texte de 74
traitement 73

F

Flux de travaux (exemple de notification) 71
Fonction BiDiBOTransformation
API de transformation bidirectionnelle 116
Fonction BiDiBusObjTransformation
API de transformation bidirectionnelle 116
Fonction BidiStringTransformation 116

G

Gestion du texte bidirectionnel 115
Gestionnaire de données 3

I

IBM Tivoli License Manager 2, 43
IDL (Interface Definition Language / langage de définition de l'interface) 30
Informations propres à l'application 63
dans un attribut d'objet métier 63
dans un objet métier 63
dans une instruction d'objet métier 64
Initialisation
collaboration 58
connecteur 67
Installation
environnement local 104
Instruction 20, 62, 74
Informations propres à l'application 64, 77
Interactions des appels 10
Interactions publication-abonnement 8, 10, 31
Interactions requête/réponse 11, 31
InterChange Server 25, 33
contrôleur de connecteur 26
haute disponibilité 27
mappes et 85
modes de 45
outil de configuration pour 44
outils de développement 45
outils pour 43
plate-forme de 25
plusieurs 6
pools de connexion à des bases de données 27
présentation 1
référentiel 26
reprise sur incident 29
service de connectivité de la base de données 26
service de gestion des événements 25
service Transactions 27
traitement simultané 78
Interface d'accès au serveur 4, 10

Interface Definition Language (IDL) 30
Interface universelle de connexion aux bases de données (Open Database Connectivity - ODBC) 26
Internet 6, 32
Interrogation 73
Intervention sur appel 11, 49
Isolement 97, 98
ITLM 2, 43

J

Java Messaging Service (JMS) 30

L

Langues
prise en charge des langues bidirectionnelles par les mappes 115
prise en charge par WebSphere 106, 112
langues bidirectionnelles
prise en charge par les mappes 115
Langues bidirectionnelles
bidirectionnalité 107
gestion du texte 115
mise en forme 109
prise en charge par WebSphere 106, 112
remarques relatives à la conception 116
traitement des données dépendantes de l'environnement local 104
transformations de la présentation 110
Les informations propres à l'application
dans une instruction d'objet métier 77
Ligature
définie 110

M

Machine virtuelle Java (JVM) 25
Map Designer 24, 46, 87
Mappage 23, 85
gestion de 24
lors de l'utilisation 23
objectif du 85
Référence de la mappe 90
transformations des relations 89
Transformations simples 88
Mappage de données
Voir Mappage
Mappes 3, 87
API de mappage 87
création 46
transformations 88
mappes pour les langues bidirectionnelles
prise en charge 115
Métadonnées 79
Métadonnées de configuration
codage des données 106
Mise en forme
langues bidirectionnelles 109

Modèle de collaboration 45, 47, 50

N

Niveau de transaction maximal 99
Niveau de transaction minimal 99
Niveau de transaction strict 100
Niveaux 35
Niveaux de sécurité 35
Niveaux de transaction 98, 100
Notification d'événement 10, 69
configuration 70
rôle de l'objet métier 19
rôle du connecteur dans la 69

O

Object Discovery Agent Development Kit (ODK) 46
Object Request Broker (ORB) 30
Objet de collaboration 47, 50
Objet métier 3, 19, 59
à plat 21
attribut 62
Attribut 21, 60
composants 20
construction 74, 78
déconstruction 78
en tant que notification d'événement 19
en tant que réponse 20
en tant que requête 19
enfant 21
générique 22, 23
hiérarchique 21
informations propres à l'application
dans 63
instruction 20, 62
mappage 23, 85
modification 64
propre à l'application 22, 62
rôles d'un 19, 20
type 20
valeur d'attribut 21
Objet métier à plat 21
Objet métier enfant 21
Objet métier générique 22, 23
Objet métier hiérarchique 21
Objet métier propre à l'application 22, 62
Objet métier réponse 20
Objet métier requête 19
ODBC (Open Database Connectivity) 26
ORB (Object Request Broker / Répartiteur de requêtes d'objets) 30

P

Port 50
Prise en charge
environnement local 103
Prise en charge des collaborations
langues bidirectionnelles 114
Prise en charge des données
bidirectionnelles
environnement local 103

- Prise en charge des langues
 - bidirectionnelles par les mappes 115
- Prise en charge des langues bidirectionnelles par WebSphere 106, 112
- Prise en charge par Adapter Framework
 - langues bidirectionnelles 113
- Process Designer 45, 47
- Processus métier
 - implémentation de 53
 - logique 3, 54
- Processus métier process
 - longue durée 49
- Propriétés
 - collaboration 47
 - connecteur 46, 82
- Propriétés standard (connecteur) 82
- Protocole IIOP (Internet Inter-ORB protocol) 31

R

- Référence de la mappe 90
- Référentiel 26
 - connectivité à 26
- Référentiel InterChange Server 26
- Relation d'identité 89
- Relation de non-identité 89
- Relation de recherche 89
- Relationship Designer 24, 46, 88
- remarques relatives à la conception
 - langues bidirectionnelles 116
- Remarques relatives à la conception
 - environnement local 105
- Reprise sur incident 29, 100

S

- Scénario 53, 56
 - étape d'activité 55, 56
 - invalidation 95
 - options d'organisation 53
 - transactionnel 92, 93
- Sécurisation 33
- sécurité 33, 34, 36, 37
- Sécurité 32, 33, 34, 35, 36, 37
- Sécurité asymétrique et dynamique 34
- Service de gestion des événements 25, 26
- Service Transactions 27
- Signe diacritique
 - défini 110
- System Manager 27, 44, 46
- System Monitor 46
- Système d'intégration de processus d'entreprise WebSphere 1

T

- Technologies de messagerie 32
- Texte bidirectionnel
 - API BiDi 116
 - gestion 115
- Traitement des données
 - dépendant de l'environnement local 104

- Traitement des requêtes 75, 78
 - à base d'instructions 76
- Transaction 28
- Transformations de la présentation
 - langues bidirectionnelles 110
- Type d'attribut composé 61
- Type d'attribut de base 61
- Types de messages 36

V

- Validation en deux phases 92
- Virtual Test Connector 45

W

- WebSphere MQ 30, 32
- WebSphere pour les langues bidirectionnelles, prise en charge 106, 112

IBM