



## Présentation technique du produit





## Présentation technique du produit

**Remarque**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations à la section Remarques à la fin du document.

**Remarque**

Les captures d'écran de ce manuel ne sont pas disponibles en français à la date d'impression.

**Deuxième édition - avril 2006**

Réf. US : SC10-4208-01

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

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 2006. Tous droits réservés.

© Copyright International Business Machines Corporation 2006. All rights reserved.

---

## Table des matières

### Avis aux lecteurs canadiens . . . . . v

### Chapitre 1. Intégration à l'entreprise . . . 1

Intégration entre entités d'une entreprise . . . . . 2

Intégration entre entreprises . . . . . 2

WebSphere Integration Developer . . . . . 3

Normes . . . . . 4

Rôle du développeur d'intégration . . . . . 4

### Chapitre 2. Architecture SCA. . . . . 7

Composants de service . . . . . 8

Objets SDO . . . . . 10

Qualificatifs de service. . . . . 11

Modules . . . . . 12

Importations et exportations. . . . . 14

Types de liaison d'importation et d'exportation

de service . . . . . 14

Sélection des liaisons appropriées . . . . . 16

Types d'implémentation de service . . . . . 16

Objets Java . . . . . 16

Processus BPEL . . . . . 17

Machines d'état . . . . . 18

Règles métier. . . . . 19

Sélecteurs . . . . . 20

Tâche manuelle . . . . . 21

Mappe d'interface . . . . . 22

Flux de médiations. . . . . 23

Références autonomes. . . . . 24

Informations connexes. . . . . 24

### Chapitre 3. Apprentissage des outils 27

Vue d'ensemble de la vue Bienvenue . . . . . 27

Tutoriels de la vue Bienvenue . . . . . 28

Exemples de la vue Bienvenue . . . . . 28

Centre de documentation. . . . . 29

### Remarques . . . . . 31



---

## Avis aux lecteurs canadiens

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

### Illustrations

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

### Terminologie

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

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

### Claviers

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








### OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

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

### Nomenclature

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

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

## Brevets

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

## Assistance téléphonique

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

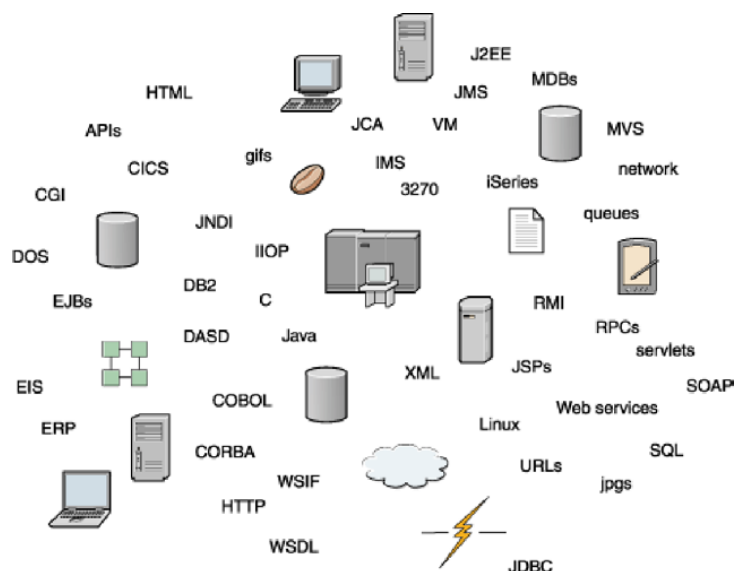


# Chapitre 1. Intégration à l'entreprise

L'intégration à l'entreprise désigne l'intégration d'applications, de données et de processus dans une entreprise ou un ensemble d'entreprises. Dans cette section, nous examinons le défi que constitue cette tâche et la solution proposée par WebSphere Integration Developer.

Vous êtes chargé de créer un portail d'accès à vos applications métier pour vos clients. Ce portail doit permettre d'accéder à des dizaines d'applications ainsi qu'à des données réparties sur toutes les entités de l'entreprise. Vous êtes aussi chargé d'ajouter les applications de vos partenaires commerciaux à ce portail. L'intégration implique également le développement de processus, puisque la séquence d'applications assemblées suit une certaine logique. L'entreprise possède 20 entités et une dizaine de partenaires commerciaux. Le portail doit être disponible sur le Web 24 heures sur 24. Votre équipe se compose de six développeurs, vous y compris, et vous disposez de quatre mois pour livrer le projet.

Face à un tel défi, bon nombre envisagerait une sorte de collage des diverses technologies apprises au fil du temps, comme suit :



Ce projet, même complexe, n'est pas irréalisable. Les difficultés principales, outre la réunion de matériels et de logiciels hétérogènes, résident dans les contraintes de temps et de ressources. Il vous faut des outils puissants pour regrouper rapidement les applications et les données éparpillées dans les entreprises concernées. Le codage manuel n'est donc pas envisageable.

Etes-vous le seul dans cette situation ? Non. Il s'agit de difficultés répandues qui se sont agrégées au fil des ans. Selon une enquête menée en décembre 2001 (*magazine CIO*), l'intégration d'applications demeure l'une des trois priorités technologiques. Selon un autre article datant de l'hiver 2001 (*The Business Integrator Journal*), deux développeurs sur trois ont recours à un logiciel d'intégration pour concevoir des solutions Web. En moyenne, chaque développeur intègre trois systèmes différents.

Examinons deux facteurs entraînant cette situation critique : l'intégration entre entités d'une entreprise et l'intégration entre entreprises. Etudions ensuite la solution proposée par WebSphere Integration Developer, en particulier l'importance de l'adhésion aux normes de l'industrie. Enfin, traitons du spécialiste de l'intégration qui utilisera les outils WebSphere Integration Developer pour résoudre les difficultés décrites précédemment.

---

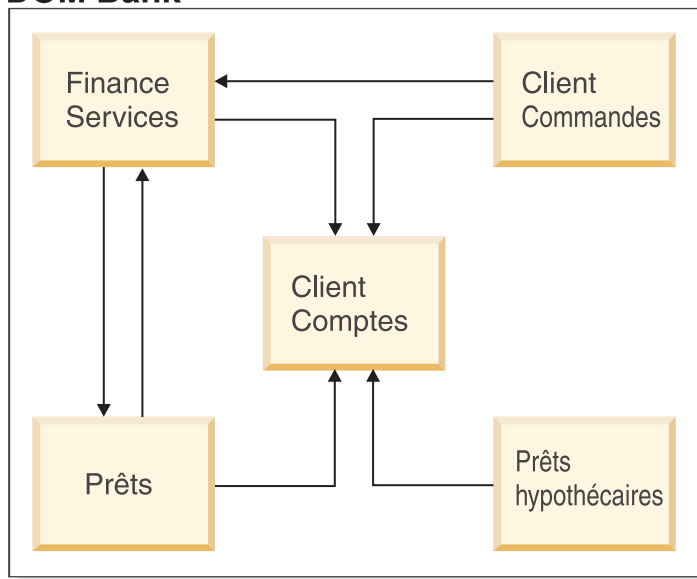
## Intégration entre entités d'une entreprise

Aujourd'hui, les entités d'une entreprise doivent souvent collaborer, et par conséquent, ont besoin d'intégrer leurs applications.

Des entités jusqu'alors indépendantes sont intégrées grâce au moyen de connexion offerts par la technologie. Par ailleurs, en termes d'efficacité, il est préférable qu'elles fonctionnent de façon plus coopérative afin de réduire les coûts et optimiser les résultats. L'objectif commercial les oblige également à se rapprocher. Les départements de marketing et de R+D souhaitent concevoir un produit rentable. En intégrant les connaissances du marché aux informations de développement du produit, la société augmente ses chances de réussir. La collaboration entre entités permet aux compagnies d'amortir les applications commerciales en y recourant dans divers contextes.

L'intégration entre entités est plus abordable que celle entre entreprises, car la sécurité est moins menacée et les interactions plus faciles à gérer. Les entités utilisent généralement les mêmes protocoles, systèmes d'exploitation et langages. En d'autres termes, il s'agit d'un environnement assez homogène. Toutefois, l'essentiel est de disposer des outils appropriés pour intégrer rapidement les applications. Dans le diagramme ci-après, la banque DOM possède plusieurs entités qui doivent se partager des informations. Il y a plusieurs années, pour faire face à ce besoin, elle imprimait des copies des informations d'une entité puis les transmettait à une autre entité, dotée de ses propres systèmes et applications. De nos jours, la banque DOM doit créer des applications intégrées qui englobent ses entités pour aller aussi vite que la concurrence.

### DOM Bank



---

## Intégration entre entreprises

Les besoins pour l'intégration d'applications entre entités s'appliquent également entre entreprises, car les associations ou les acquisitions requièrent le partage de données et de processus.

La technologie permet aux entreprises d'être connectées dans des domaines utiles. Par exemple, un fabricant automobile peut établir un processus intégré avec un fournisseur de pneus pour que ce dernier soit automatiquement informé des éventuelles ruptures de stock. L'intégration entre les entreprises est donc motivée par des facteurs économiques. Les liens étroits entre sociétés réduisent les délais d'attente et les coûts impliqués. Ces processus automatisés permettent d'accélérer le traitement des transactions

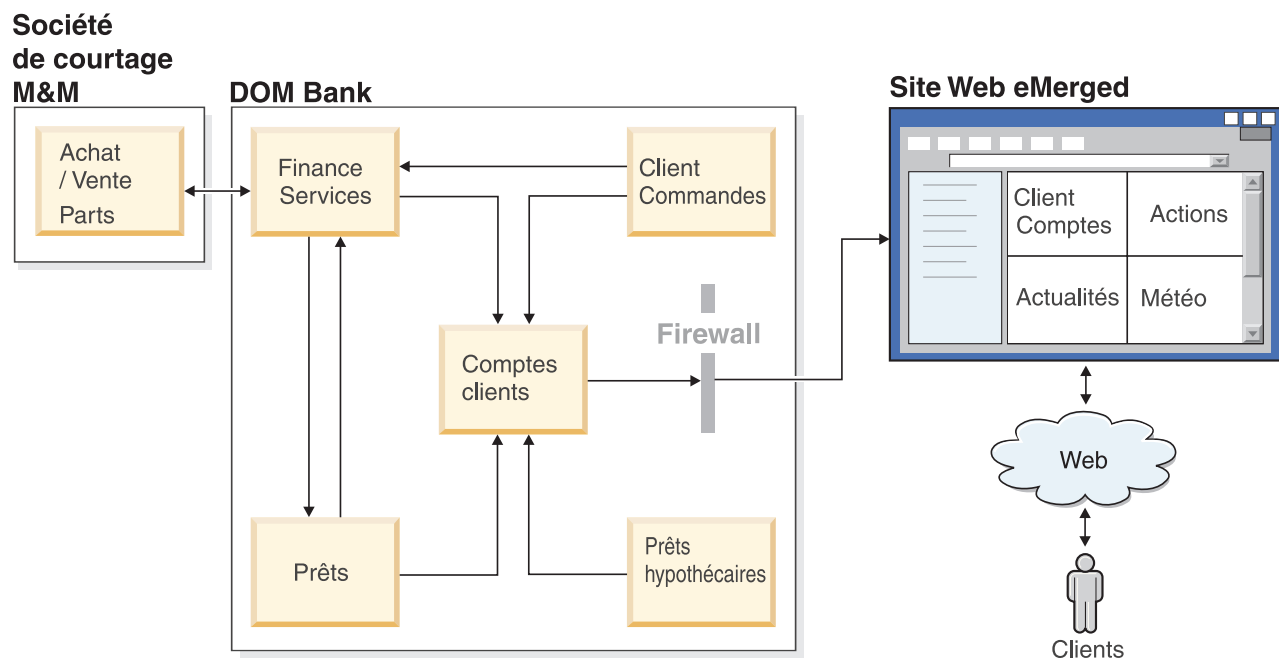
interentreprises, de diminuer les frais de déplacement et de réduire les temps de réunion en face-à-face. Les frais d'administration sont également inférieurs et les délais de notification, de livraison et de facturation sont améliorés.

Cependant, à chaque société son histoire. Les applications sont codées dans différents langages, sur différentes plateformes et avec des protocoles de communication propres. Par ailleurs, les risques en matière de sécurité sont bien réels. Quels que soient les besoins et avantages d'une telle intégration, le coût de son établissement peut prendre des proportions considérables sans les outils adéquats.

## WebSphere Integration Developer

WebSphere Integration Developer apporte une solution aux défis d'intégration auxquels l'entreprise fait face quotidiennement. Ce logiciel est conçu comme un environnement de développement d'intégration complet pour les créateurs d'applications intégrées. Pour faciliter et accélérer le développement d'applications intégrées, cet environnement propose une couche d'abstraction qui sépare de l'implémentation sous-jacente les composants à représentation visuelle que vous gérez.

Les applications intégrées sont complexes. Elles peuvent appeler des applications situées sur des systèmes d'information d'entreprise (EIS), impliquer des processus métier interservices ou interentreprises et appeler des applications écrites en local ou à distance dans plusieurs langues et exécutées sur divers systèmes d'exploitation. A titre d'exemple, eMerged Corporation provient de la fusion de la banque DOM et de la société de courtage M&M Discount. Cette fusion comporte tous les éléments cités précédemment : des applications situées sur des systèmes EIS, des processus métier et des applications appartenant aux deux anciennes sociétés devaient être partagés et présentés de manière homogène au nouvel ensemble de clients. eMerged a accompli cette tâche. Comme illustré dans le diagramme ci-après, les clients des deux anciennes entreprises peuvent désormais accéder en ligne à toutes leurs informations financières.



eMerged créait des applications intégrées à usage interne et pour ses clients à l'aide des outils WebSphere Integration Developer (WID). Ces outils représentent les applications (y compris les applications distantes, situées sur des systèmes EIS) et les processus métier sous la forme de composants. Ces composants sont créés puis assemblés dans d'autres applications intégrées (c'est-à-dire des applications créées à partir d'un ensemble de composants), à l'aide d'éditeurs visuels qui insèrent une couche

d'abstraction entre les composants et leurs implémentations. Doté de ces outils, le développeur peut créer une application intégrée sans connaître en détail l'implémentation sous-jacente de chaque composant.

Pour la création d'une application intégrée, ces outils permettent une approche descendante dans laquelle l'implémentation d'un ou de plusieurs composants n'existe pas et est ajoutée ultérieurement. Ils offrent également une approche ascendante dans laquelle les composants sont déjà implémentés : le développeur les assemble par déplacement de la souris dans un éditeur visuel et crée un flux logique entre eux en les reliant avec des traits. Un environnement de débogage et de test implique la réalisation de tests complets avant le déploiement des applications vers un serveur de production. La définition de points de surveillance permet de visualiser en temps réel le mode d'utilisation d'une application pour affiner son réglage afin d'obtenir des performances optimales.

Les outils WebSphere Integration Developer sont fondés sur une architecture orientée service. Les composants sont des services et une application intégrée impliquant de nombreux composants est un service. Les services créés sont conformes aux principales normes de l'industrie. Les processus métier, désormais des composants, sont créés de la même façon à l'aide d'outils visuels faciles à utiliser conformes au langage standard BPEL (Business Process Execution Language). WebSphere Integration Developer est disponible sur les plateformes Windows et Linux.

Les outils WebSphere Integration Developer offrent notamment les avantages suivants :

- leur apprentissage est facile ;
- ils sont applicables à des situations d'intégration complexes ;
- vous pouvez générer rapidement des applications conformes aux normes de l'industrie.

## Normes

Les applications créées avec WebSphere Integration Developer sont conformes aux normes de l'industrie associées à l'architecture orientée service.

Personne ne souhaite créer d'application liée à un code propriétaire dont la prise en charge n'est plus assurée après quelques années ou qui entraîne des frais de licence onéreux. Par conséquent, l'intégration normalisée constitue un aspect fondamental de WebSphere Integration Developer. La connectivité utilise des normes JCA (J2EE Connector Architecture). La messagerie asynchrone, souvent employée dans les grandes applications où la livraison garantie des données est nécessaire, utilise la norme JMS (Java Message Service). WebSphere Integration Developer peut facilement intégrer des services Web à l'aide du protocole SOAP (Simple Object Access Protocol). La description d'un service utilise la norme WSDL (Web Services Description Language), bien établie. La définition d'un processus métier emploie la norme BPEL (Business Process Execution Language).

Ces interfaces et composants normalisés comprennent une architecture extensible et connectable. Toutefois, les éléments propriétaires ne sont pas exclus : ils sont accessibles par le biais d'interfaces normalisées. Cela signifie que des applications créées dans WebSphere Integration Developer peuvent interagir, par exemple, avec des applications .NET. Cette section sur l'architecture propose un lien vers la section sur l'architecture SCA (Service Component Architecture) qui fournit une liste exhaustive des normes prises en charge.

## Rôle du développeur d'intégration

Le développeur d'intégration est l'utilisateur principal de WebSphere Integration Developer. A l'aide d'outils visuels, il peut créer une application intégrée complexe, sans connaissance approfondie de l'implémentation sous-jacente.

WebSphere Integration Developer présente les applications et les processus métier en tant que *composants*. L'implémentation des composants reste masquée et les composants interagissent par le biais d'interfaces. Par conséquent, les développeurs d'intégration n'ont pas besoin de connaître en détail l'implémentation sous-jacente des composants pour créer une application intégrée qui utilise ces composants. Toutefois, ils

doivent posséder de vastes connaissances techniques dans le domaine de l'intégration, car une certaine compréhension des systèmes EIS, des processus métier et des applications codées en Java ou dans d'autres langages est nécessaire. Par exemple, un architecte peut avoir une bonne compréhension du mode de fonctionnement d'un système sans connaître en détail les fonctions de chaque composant. A l'instar de l'architecte, le développeur d'intégration est, dans l'entreprise, le concepteur de l'application globale, tandis que d'autres collaborateurs codent l'implémentation des composants spécifiques.



---

## Chapitre 2. Architecture SCA

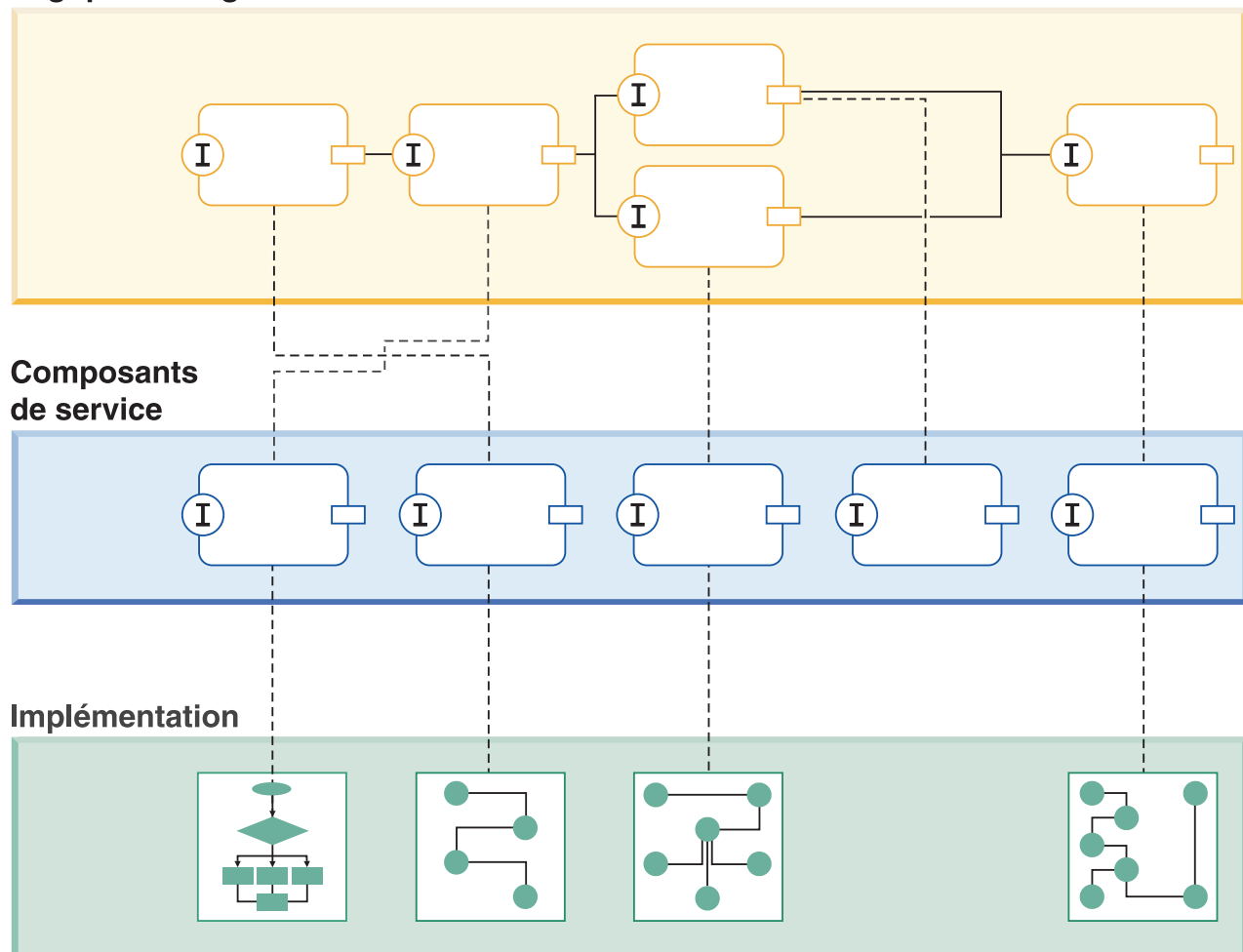
L'architecture SCA (Service Component Architecture), basée sur l'architecture orientée service conforme aux normes de l'industrie, présente tous les processus métier (par exemple, les services Web, les ressources de service EIS, les flux de travaux ou les bases de données) dans un mode orienté service. Dans cette section, nous examinons de manière générale les services et les objets de données de service créés par cette architecture qui forment ensemble la logique métier et concernent les données métier.

L'architecture SCA a pour but de séparer la logique d'intégration métier de l'implémentation de façon à ce que le développeur d'intégration puisse se concentrer sur l'assemblage d'une application intégrée plutôt que sur les détails de l'implémentation. Pour ce faire, les composants de service contenant l'implémentation des différents services requis par les processus métier sont créés. Le résultat obtenu est une architecture à trois couches (logique d'intégration métier, composants de service et implémentation), comme illustré dans le diagramme ci-dessous.



Les composants de service contiennent l'implémentation. Le développeur d'intégration peut donc les assembler graphiquement sans connaître les détails de l'implémentation. Les composants de service permettent également au développeur d'intégration, ou à son collaborateur, d'ajouter l'implémentation ultérieurement. Comme illustré dans le diagramme du produit, les composants sont assemblés visuellement. En d'autres termes, vous n'êtes pas exposé au code contenu dans les composants. Dans le niveau de logique métier présenté dans le diagramme ci-après, les composants sont assemblés indépendamment de leur implémentation. L'architecture orientée service vous permet donc de concentrer vos efforts sur la résolution des questions métier en utilisant et en réutilisant des composants sans vous préoccuper de la technologie d'implémentation des services employés.

## Logique d'intégration métier



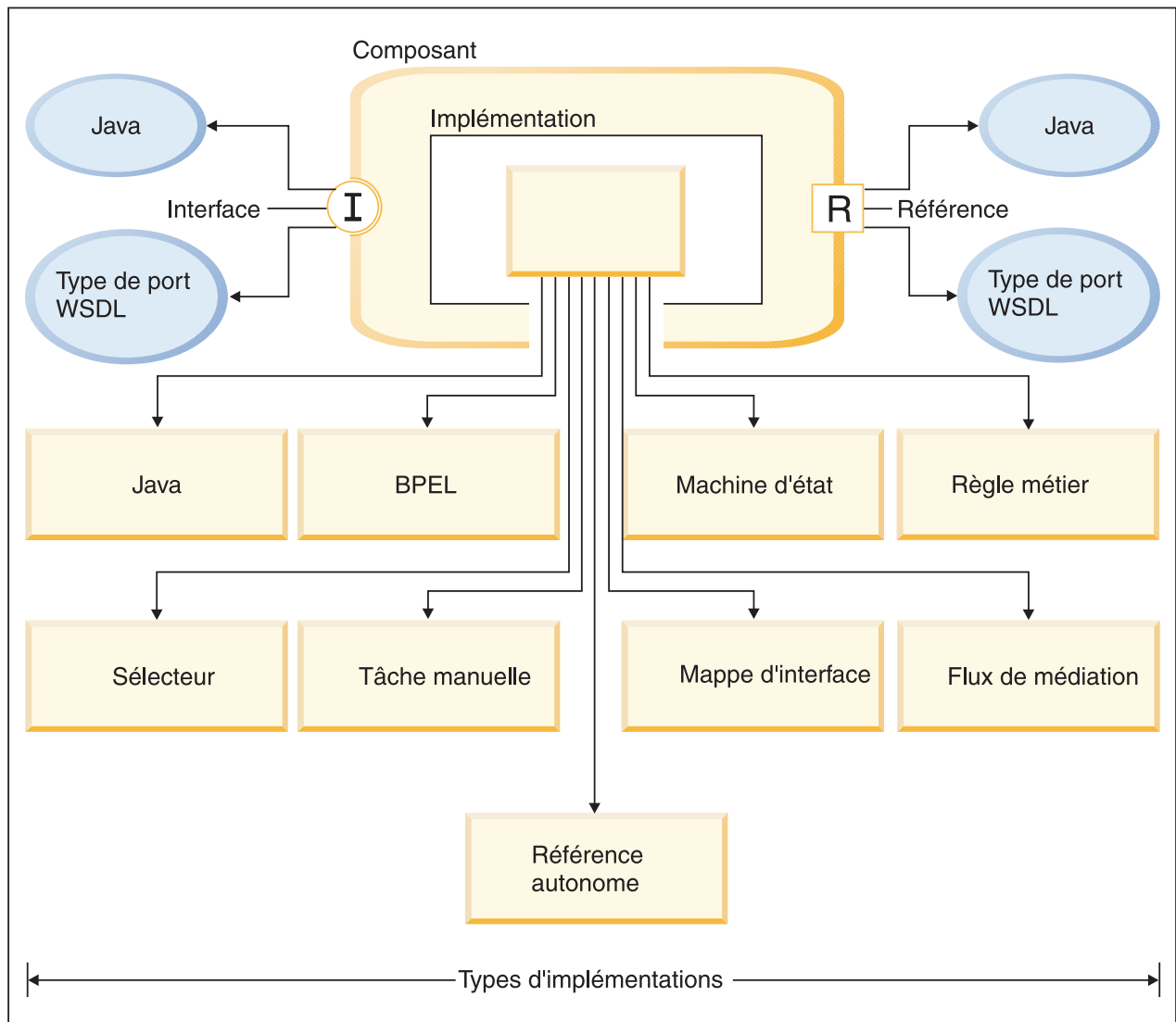
## Composants de service

Un composant de service configure une implémentation de service. Il est présenté dans un schéma de principe standard.

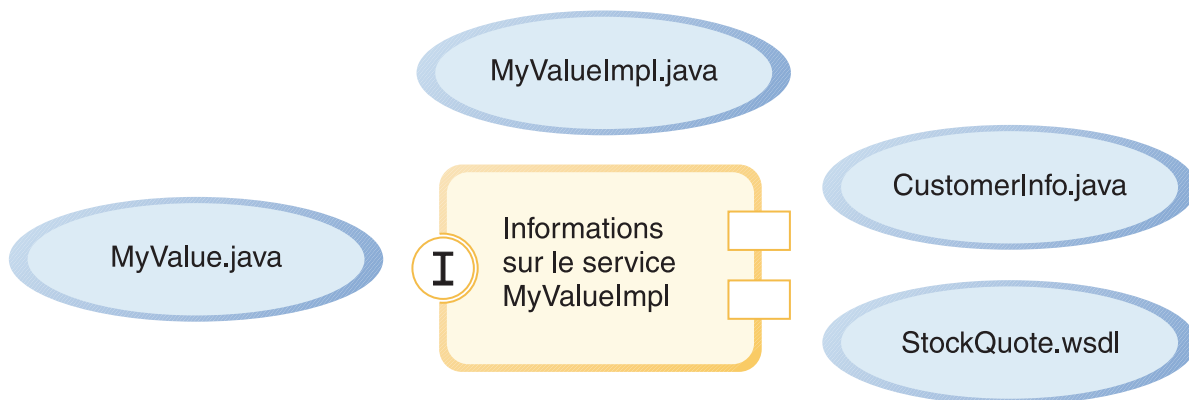
Un composant est constitué d'une implémentation (qui est masquée lors de l'utilisation des outils WebSphere Integration Developer), d'une ou de plusieurs interfaces (qui définissent les entrées, les sorties et les échecs de cette implémentation) et de zéro ou plusieurs références. Une référence identifie l'interface d'un autre service ou composant que ce composant requiert ou utilise. Une interface peut être définie dans un des deux langages suivants : un type de port WSDL ou Java. Une interface prend en charge les interactions de style synchrone et asynchrone. L'implémentation d'un composant peut s'effectuer dans diverses langues.

Une interface peut être de type WSDL ou Java. Toutefois, plusieurs interfaces n'acceptent pas la combinaison des deux, WSDL et Java.



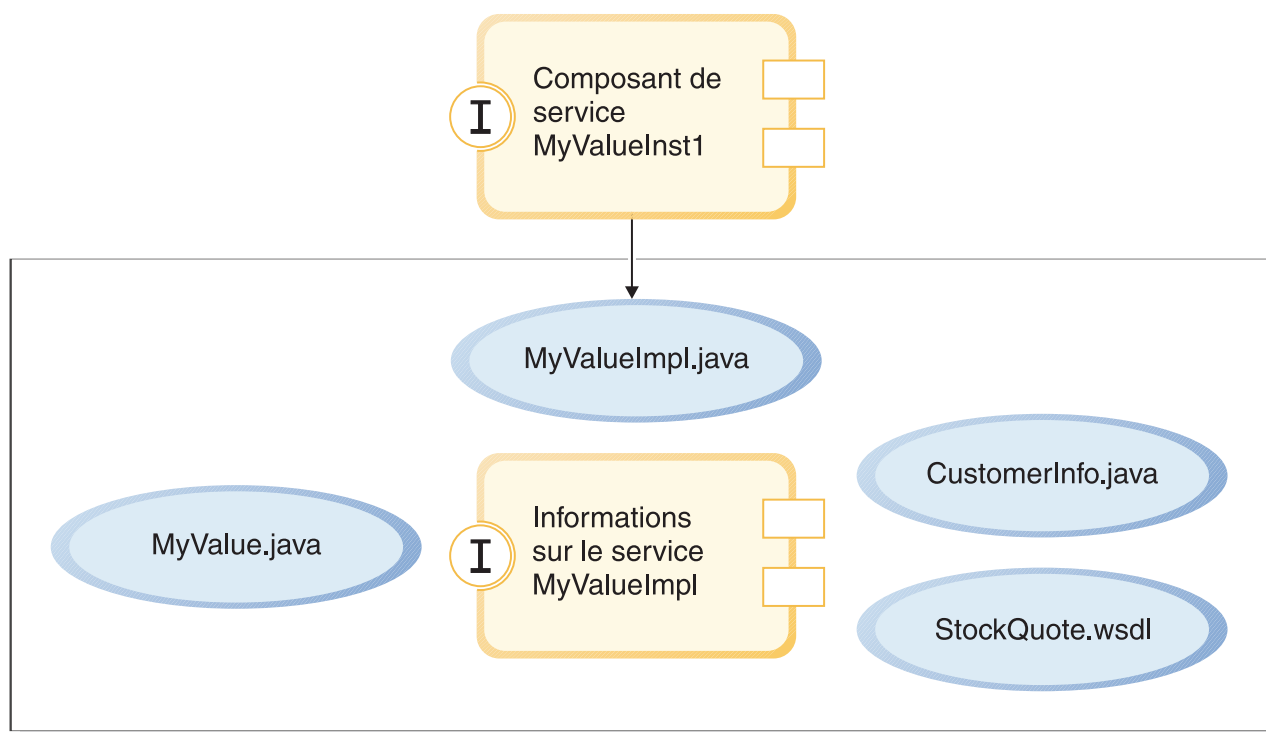


Un composant est représenté au centre de l'illustration ci-après. Son implémentation, `MyValueImpl`, comme son interface, est en Java. Il comporte deux références : une autre interface Java et une interface WSDL.



Lorsque vous gérez ce composant (voir ci-après), vous ne visualisez en réalité que celui-ci. Lorsque le composant est cité en référence par un autre, un trait s'affiche jusqu'à son interface. Lorsqu'il fait référence à un autre composant, un trait part de son point de référence jusqu'à l'interface de l'autre composant. Une référence représente un service utilisé par ce composant. En nommant une référence et en spécifiant uniquement son interface, l'auteur de l'implémentation du composant peut différer à une date ultérieure l'établissement de la liaison faisant référence à un service réel. Le spécialiste de l'intégration effectuera cette opération à cette date en connectant la référence à l'interface d'un autre composant ou d'une autre importation. Ce couplage lâche, qui permet l'établissement différé des liaisons et la réutilisation des implémentations, est une des principales raisons justifiant l'utilisation de l'architecture SCA (Service Component Architecture) de WebSphere Integration Developer.

Un composant peut également être doté de propriétés et de qualificatifs. Un qualificatif est une directive de qualité de service (QoS) applicable aux interfaces et aux références pour l'environnement d'exécution.

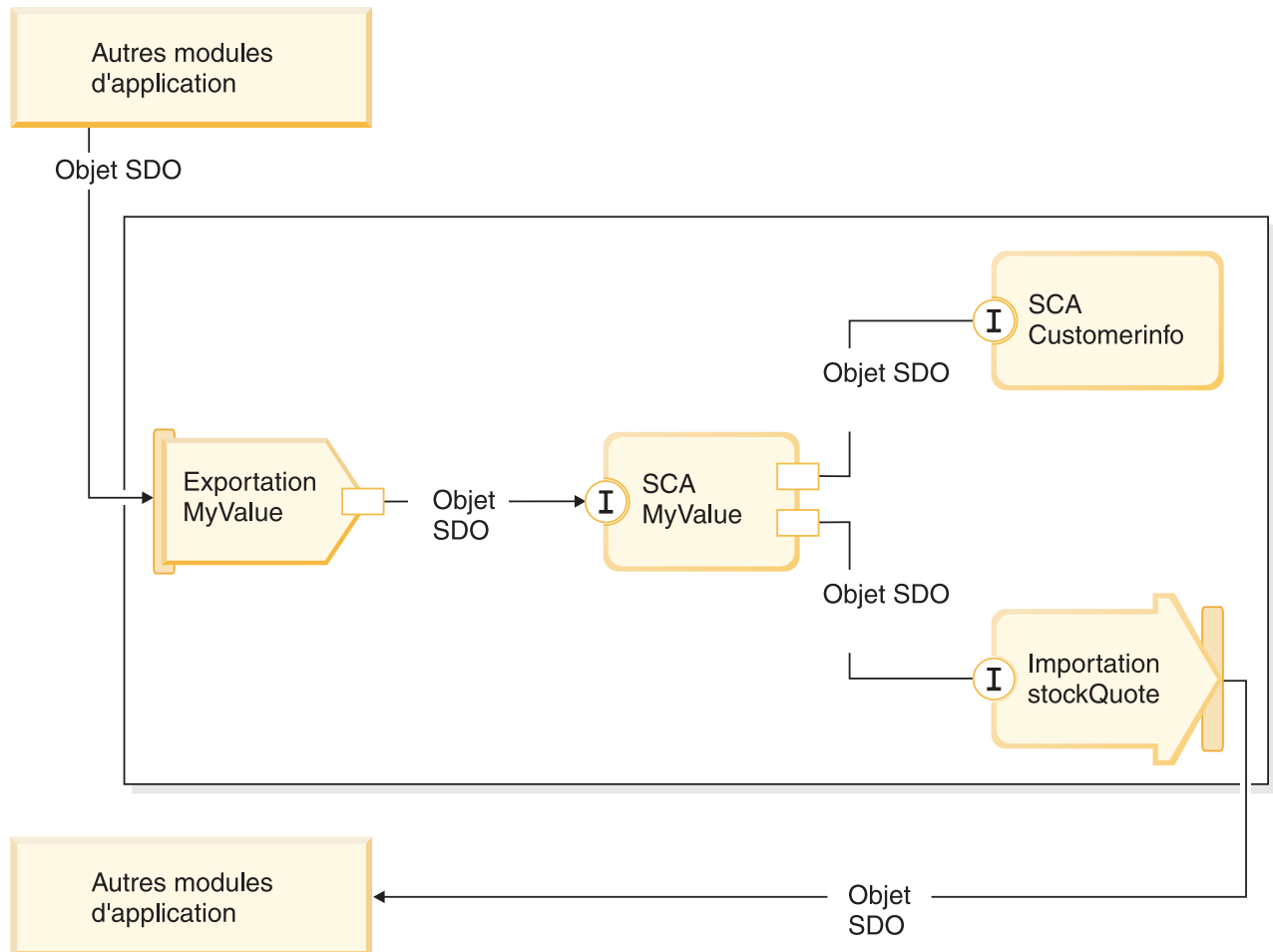


## Objets SDO

Les objets SDO (Service Data Object) apportent un complément à l'architecture SCA (Service Component Architecture). Cette architecture définit les services en tant que composants ainsi que leurs interconnexions. Les *objets SDO* définissent les données qui circulent entre ces composants.

Chaque composant transmet les informations comme paramètres d'entrée et de sortie. Lors de l'appel d'un service, les objets de données sont transmis sous la forme d'un document XML utilisant un codage littéral lors de l'utilisation d'un type de port WSDL ou sous la forme d'un objet Java lors de l'utilisation d'une interface Java. Les objets de données constituent le format privilégié des données et des métadonnées des services SCA. Comme pour les composants, les objets SDO séparent l'objet de données de son implémentation. Par exemple, un composant peut interagir avec des bons de commande alors que ceux-ci utilisent, par exemple, une interface JDBC ou EJB pour la mise à jour des données. Les objets SDO permettent au développeur d'intégration de concentrer ses efforts sur la gestion des artefacts métier. En fait, les objets de données de service sont transparents pour le développeur d'intégration. Ils sont définis par une JSR (Java Specification Request) d'objets de données de service.

Dans le diagramme ci-dessous, les objets de données de service sont transmis d'un service externe à une exportation, d'une exportation à un composant, d'un composant à un composant, d'un composant à une importation et d'une importation à un service. Les importations et exportations sont présentées dans une autre section intitulée Importations et exportations.



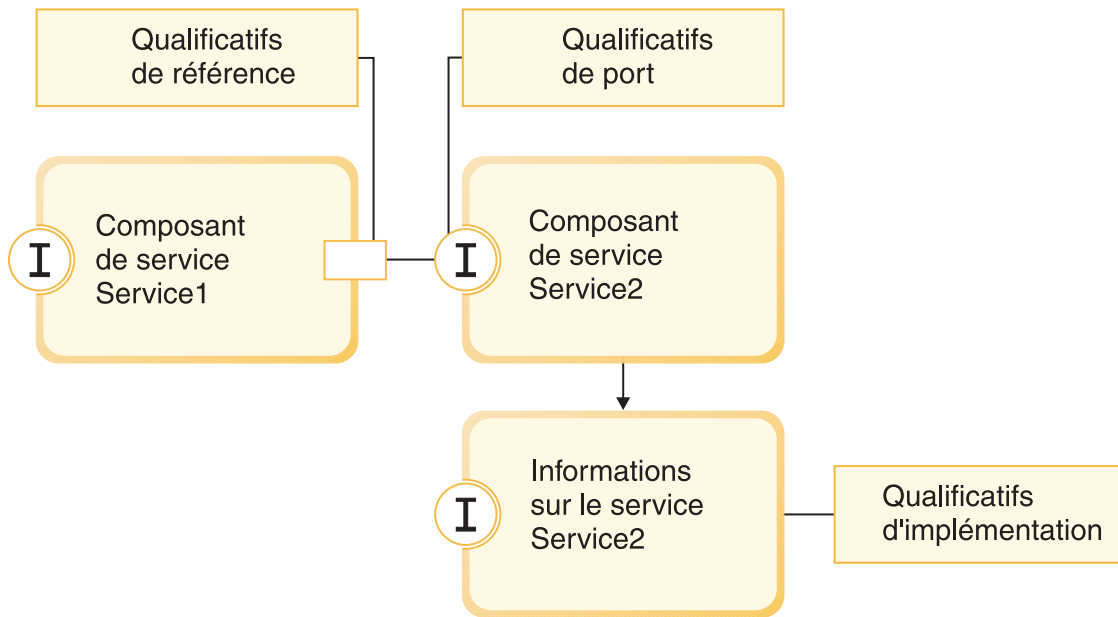
## Qualificatifs de service

Une application communique ses besoins en qualité de service (QoS) à l'environnement d'exécution par la spécification de *qualificatifs de service*. Ces qualificatifs régissent l'interaction entre un client de service et un service cible.

Les qualificatifs peuvent s'appliquer à des références de composant de service, à des interfaces et à des implémentations. La déclaration des valeurs QoS est indépendante de l'implémentation. Par conséquent, vous pouvez modifier ces valeurs sans modifier l'implémentation ou les configurer différemment lorsque plusieurs instances de la même implémentation sont employées dans divers contextes.

Les catégories de qualificatif sont les suivantes :

- transaction (règles pour le type de transaction) ;
- session d'activité (règles de connexion à la session active) ;
- sécurité (règles de permission) ;
- fiabilité asynchrone (règles pour la livraison asynchrone des messages).



## Modules

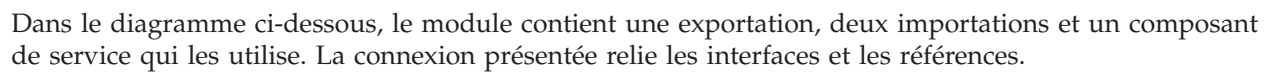
Un *module* est une unité de déploiement qui détermine les artefacts fournis dans un fichier EAR (Enterprise Archive). Les composants d'un module sont situés au même endroit pour des raisons de performances et peuvent transmettre leurs données par référence. Un module peut être considéré comme un mécanisme de mesure de la portée, c'est-à-dire qu'il définit une limite organisationnelle pour les artefacts.

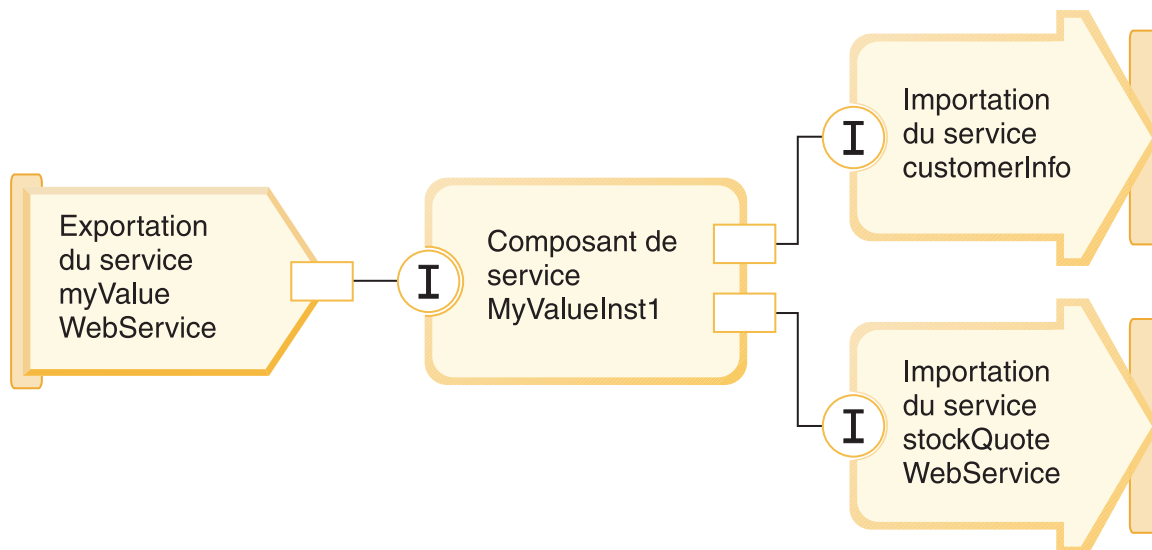
Un module est un fichier composite de composants de service, d'importations et d'exportations. Les composants de service, les importations et les exportations résident dans le même projet et le même dossier racine, qui contiennent également les connexions reliant les composants ainsi que les liaisons nécessaires pour les importations et les exportations. Un module peut également inclure les implémentations et interfaces référencées par ses composants, importations et exportations, ou celles-ci peuvent être placées dans d'autres projets, tels qu'un projet de bibliothèque.

Il existe deux types de modules. Premièrement, un module désigné *module* (parfois appelé module d'intégration métier) qui contient de nombreux types de composants, souvent utilisés pour prendre en charge un processus métier. Deuxièmement, un module appelé *module de médiation*, qui contient au maximum un composant, un composant de flux de médiations ainsi que zéro ou plusieurs composants Java étendant le composant de flux de médiations.

Pourquoi existe-t-il deux types de modules ? Le premier type de module est principalement conçu pour les processus métier. Un module de médiation est semblable à une passerelle vers des services externes existants, qui est couramment utilisée dans les architectures Enterprise Service Bus. Ces exportations ou services externes sont utilisés au sein d'un module de médiation par des importations ou des fournisseurs de services. Grâce à la dissociation des demandeurs de services client et des fournisseurs de services par un flux de médiations, la flexibilité et la souplesse de vos applications sont améliorées, ce qui correspond à un objectif de l'architecture orientée service. Par exemple, votre flux de médiations peut consigner des messages entrants, acheminer des messages vers un service spécifique déterminé lors de l'exécution ou transformer des données pour qu'elles puissent être transmises à un autre service. Ces fonctions peuvent être ajoutées et modifiées au fil du temps sans modification des services du demandeur ou du fournisseur.

Les implémentations, les interfaces, les objets métier, les mappes d'objets métier, les rôles, les relations et d'autres artefacts doivent souvent être partagés par des modules. Une *bibliothèque* est un projet utilisé pour stocker ces ressources partagées.





Les artefacts du module et du module de médiation comprennent les éléments suivants :

- Une définition de module identifiant le module
- Des composants de service (c'est-à-dire des définitions des services contenus dans le module). Un nom de composant de service contenu dans un module est unique. Toutefois, un composant de service peut avoir un nom d'affichage arbitraire, qui est généralement plus utile pour les utilisateurs.
- Des importations, c'est-à-dire des définitions d'importations, qui correspondent à des appels adressés vers des services externes à ce module. Les importations comportent des liaisons, qui sont présentées dans la section Importations et exportations.
- Des exportations, qui correspondent à des définitions d'exportations, qui sont utilisées pour présenter des composants à des appelants externes à ce module. Les exportations comportent des liaisons, qui sont présentées dans la section Importations et exportations.
- Des références d'un composant à un autre dans le module.
- Des références autonomes renvoyant à des applications non définies comme des composants SCA (par exemple, JavaServer Pages), qui permettent à ces applications d'interagir avec les composants SCA. Il ne peut exister qu'un seul artefact de références autonomes par module.
- D'autres artefacts - notamment des fichiers WSDL, des classes Java, des fichiers XSD et des processus BPEL.

## Importations et exportations

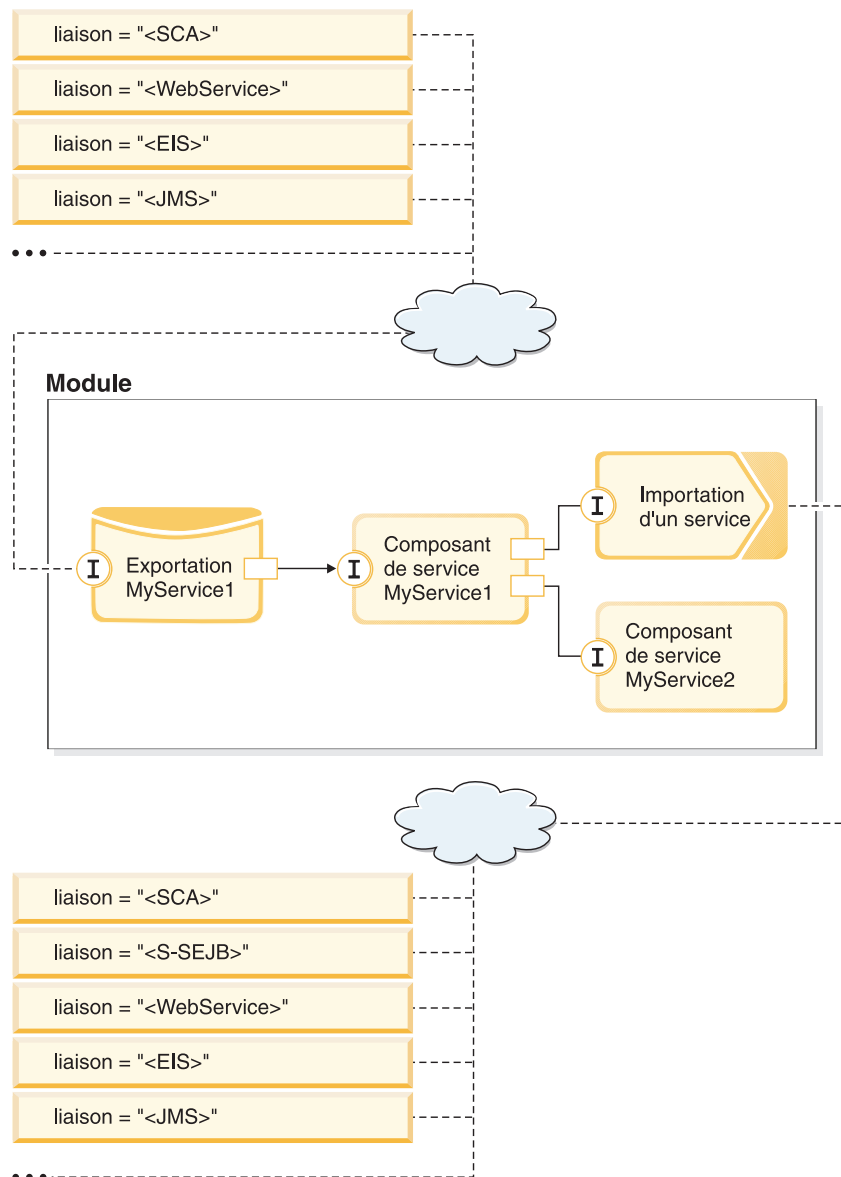
Les importations et les exportations définissent les interfaces ou points d'accès externes d'un module. Les *importations* identifient les services qui se trouvent à l'extérieur d'un module pour les rendre appelables à partir de celui-ci. Les *exportations* permettent aux composants de mettre leurs services à la disposition de clients externes. Des informations de liaison sont nécessaires pour les importations et les exportations. Plusieurs liaisons sont disponibles et des indications vous aident à choisir le type le plus approprié pour votre application.

## Types de liaison d'importation et d'exportation de service

Les importations et exportations requièrent des informations de liaison qui indiquent le mode selon lequel les données sont transférées à partir des modules. Une *liaison d'importation* décrit le mode selon lequel un service externe est lié à un composant d'importation. Une *liaison d'exportation* décrit le mode selon lequel les services d'un module sont mis à la disposition des clients.

La liaison SCA (ou liaison par défaut) permet à votre service de communiquer avec des services d'autres modules. Une importation dotée d'une liaison SCA permet d'accéder à un service d'un autre module. Une exportation dotée d'une liaison SCA permet d'offrir un service à d'autres modules. Une liaison d'importation de service Web permet de lier un service Web externe à une importation. Une liaison d'exportation de service Web permet de proposer un service en tant que service Web à des clients externes. L'assistant de reconnaissance des services d'entreprise crée les importations et les exportations qui représentent un service sur un système EIS. Les liaisons créées sont de type EIS ou JMS (Java Message Service). Une liaison EIS assure une communication synchrone avec le service situé sur le système EIS. Une liaison JMS est généralement utilisée dans des interactions avec des grands systèmes EIS où la communication asynchrone via des files d'attente de messages est essentielle pour garantir la fiabilité. Une importation (contrairement à une exportation) peut également être dotée d'une liaison EJB de session sans état.

L'éditeur d'assemblage répertorie les liaisons prises en charge et facilite leur établissement lors de la création d'une importation ou d'une exportation. Dans l'éditeur d'assemblage, une vue des propriétés affiche les informations de liaison de toute importation ou exportation.



## Sélection des liaisons appropriées

Cette section vous aide à déterminer la liaison répondant le mieux aux besoins de votre application.

Les liaisons disponibles dans WebSphere Integration Developer offrent une variété de choix. La liste ci-dessous vous permet d'identifier un type de liaison plus adapté aux exigences de votre application qu'un autre.

Envisagez d'utiliser une liaison *SCA* lorsque les facteurs suivants s'appliquent :

- Tous les services sont contenus dans les modules WebSphere Integration Developer et il n'existe par conséquent pas de service externe.
- Les performances sont importantes.
- Les modules sont à configuration groupée.

Envisagez d'utiliser une liaison de *services Web* lorsque les facteurs suivants s'appliquent :

- Vous devez accéder à un service externe via Internet ou fournir un service par ce même biais.
- Les services sont à configuration dispersée.
- Le protocole d'un service externe auquel vous accédez ou d'un service que vous souhaitez fournir est SOAP/HTTP ou JMS/HTTP.

Envisagez d'utiliser une liaison *EIS* lorsque les facteurs suivants s'appliquent :

- Vous devez accéder à un service d'un système EIS à l'aide d'un adaptateur de ressource.
- Les performances sont plus importantes que la fiabilité, ce qui signifie que les données sont transmises de façon synchrone, plutôt qu'asynchrone.

Envisagez d'utiliser une liaison *JMS* lorsque les facteurs suivants s'appliquent :

- Vous devez accéder à un système de messagerie.
- Les services sont à configuration dispersée.
- La fiabilité est plus importante que les performances, ce qui signifie que les données sont transmises de façon asynchrone, plutôt que synchrone.

Envisagez d'utiliser une liaison de *EJB de session sans état* lorsque les facteurs suivants s'appliquent :

- La liaison est destinée à un service importé qui est lui-même un EJB.
- Le service importé est à configuration dispersée.
- L'état de l'EJB n'a pas d'importance.

---

## Types d'implémentation de service

Les types d'implémentations de service correspondent aux implémentations des composants de service.

Cette section décrit les implémentations standard des services. Celles-ci apparaissent dans les services dans l'éditeur d'assemblage ou dans les processus BPEL.

## Objets Java

Une implémentation d'un composant dans Java est appelée "objet Java".

Une implémentation standard est un composant écrit en Java. Cette implémentation est parfois surnommée objet POJO ("plain old Java object"). En général, elle comporte une interface WSDL, mais elle admet également une interface Java. En présence de plusieurs interfaces, vous ne pouvez pas combiner des interfaces WSDL et Java. Toutefois, vous pouvez "relier" une application créée avec un ensemble d'interfaces WSDL à une application dotée d'un ensemble d'interfaces Java. La procédure correspondante est décrite dans l'exemple répertorié dans la galerie d'exemples de la vue Bienvenue.



Lorsque vous gérez un objet Java, le code reste masqué dans le contexte des éditeurs.

Un objet Java peut être utilisé dans un module de médiation. Il peut être déployé sur un serveur WebSphere Process Server ou WebSphere Enterprise Service Bus.

## Processus BPEL

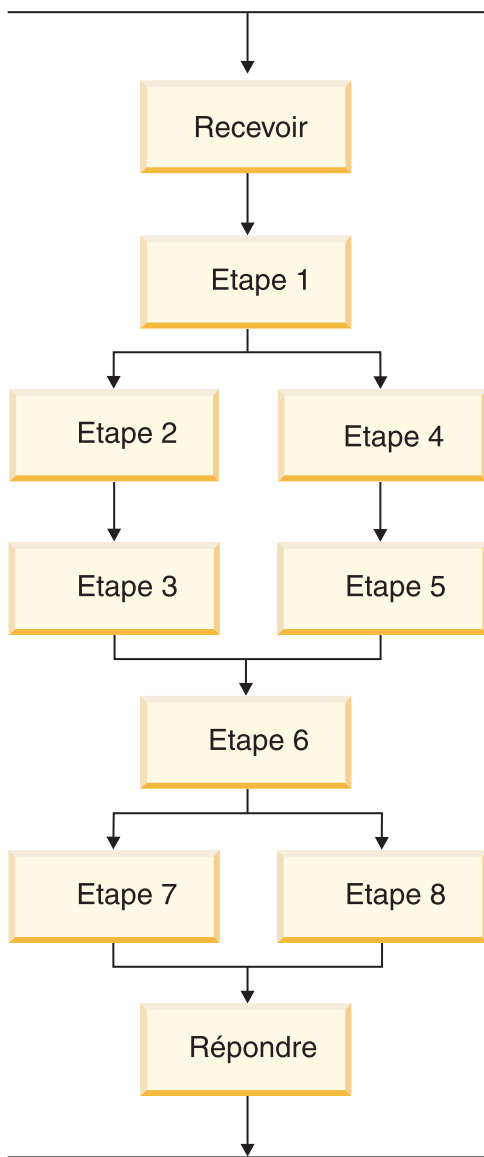
Un composant de type *processus BPEL* implémente un processus métier.

Son langage d'implémentation est le langage BPEL4WS Business Process Execution Language for Web Services standard et ses extensions IBM. Un processus BPEL implémente un service pouvant être de longue durée grâce à l'utilisation de services plus élémentaires. Un processus BPEL créé dans l'éditeur de processus peut exécuter les actions suivantes :

- décrire l'orchestration d'autres services à l'aide de graphiques de flux de commande ;
- utiliser des variables pour conserver l'état des processus ;
- employer une procédure complexe de traitement des erreurs avec gestion des échecs ;
- prendre en charge les événements asynchrones ;
- corréler les demandes entrantes avec l'instance appropriée d'un processus, par l'emploi d'ensembles de corrélation pour marquer les données métier dans la demande qui identifie l'instance (par exemple, un ID client) ;
- fournir des transactions étendues à partir d'un support de compensation complexe.

Outre ces éléments BPEL standard, WebSphere Integration Developer étend également la fonction BPEL pour inclure des personnes dans un processus qui prend en charge les *tâches manuelles*. Par exemple, cette extension peut ajouter à un processus une exigence stipulant qu'un prêt doit être approuvé par une personne.

L'éditeur de processus utilise des représentations visuelles de constructions BPEL pour vous permettre de créer rapidement et facilement votre processus métier.

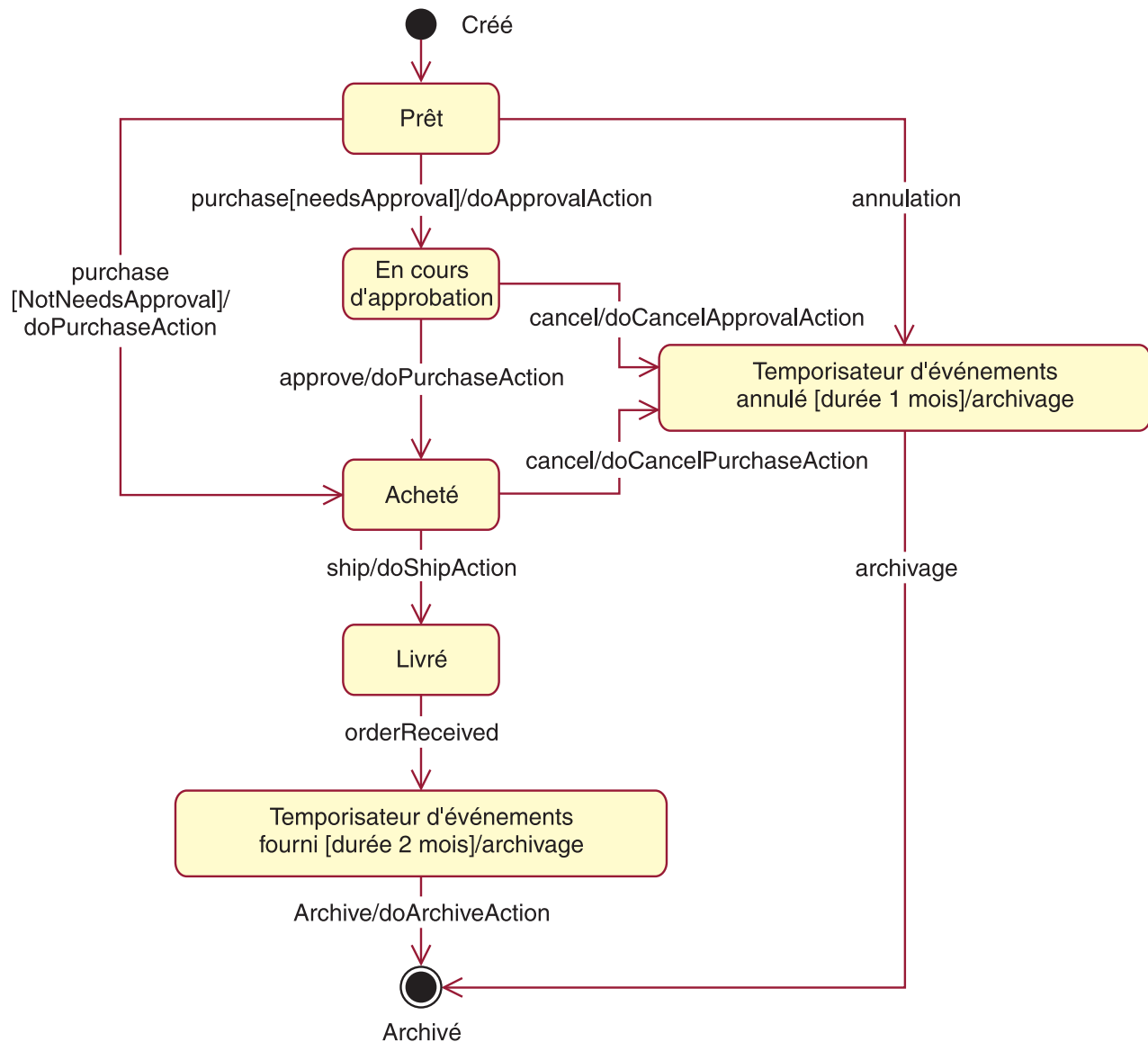


Un processus BPEL ne peut pas être utilisé dans un module de médiation. Il ne peut être déployé que sur un serveur WebSphere Process Server.

## Machines d'état

Une machine d'état représente un autre moyen de créer un processus métier. Elle est adaptée aux processus liés à des changements d'états plutôt qu'à un flux de contrôle. Un état définit la fonction réalisable par un artefact à un instant donné. Une *machine d'état* est une implémentation de cet ensemble d'états.

Les machines d'état constituent une méthode courante pour présenter un ensemble d'états interreliés dans un processus. Un distributeur de boissons est une machine d'état familière. Vous mettez des pièces dans la machine qui, avec la boisson attendue, rend l'appoint : la machine analyse mécaniquement les pièces à renvoyer en fonction des pièces insérées. Le diagramme ci-dessous illustre une machine d'état standard créée par l'éditeur de machine d'état. Dans la machine d'état, un article est acheté et livré à un client.



Une machine d'état ne peut pas être utilisée dans un module de médiation. Elle ne peut être déployée que sur un serveur WebSphere Process Server.

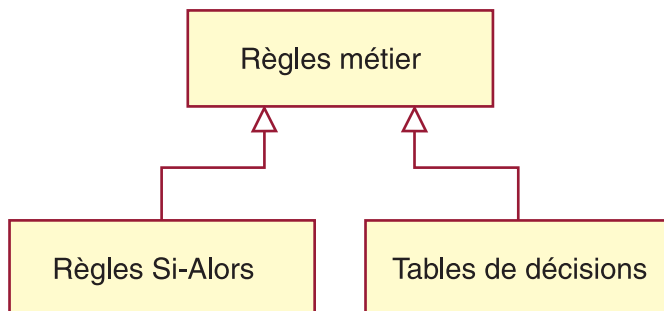
## Règles métier

Les règles métier complètent les processus métier et les machines d'état. Si une condition est associée à une variable, par exemple, une *règle métier* peut changer la valeur de cette variable lors de l'exécution. Créée par un langage de programmation visuel, une règle métier prend une décision en fonction du contexte. La décision peut être simple ou complexe. Les règles métier sont non procédurales et peuvent être modifiées indépendamment d'une application.

Les règles métier déterminent le résultat d'un processus selon un contexte. Elles sont utilisées dans des situations métier quotidiennes pour prendre une décision en fonction d'un ensemble déterminé de circonstances. Cette décision peut nécessiter de nombreuses règles pour couvrir toutes les circonstances. Les règles d'un processus métier permettent aux applications de répondre rapidement aux conditions métier qui sont en constante évolution. Par exemple, dans une société d'assurance, la règle métier d'acceptation d'une demande d'assurance auto peut être définie comme suit : *Si le candidat est de sexe*

*masculin, âgé de plus de 25 ans, propriétaire d'un véhicule de la catégorie Sport et qu'il a souscrit un contrat d'assurance auprès de la compagnie au cours des cinq dernières années, accepter sa candidature moyennant une cotisation mensuelle de 100 euros.*

WebSphere Integration Developer offre plusieurs approches pour créer des règles métier. Vous pouvez créer des règles Si-Alors ou des tables de décision ; tous ces éléments façonneront le résultat de votre processus. Ces règles sont indépendantes du processus et, par conséquent, modifiables sans devoir recréer celui-ci. Par exemple, selon l'implantation de l'entreprise, une règle peut indiquer : *Si la date est comprise entre le 26 décembre et le 1er janvier, alors offrir une remise de 20 % au titre de la rentrée.* Toutefois, en cas d'augmentation trop lente du chiffre d'affaires, vous pouvez à tout moment faire passer cette remise à 40 %.

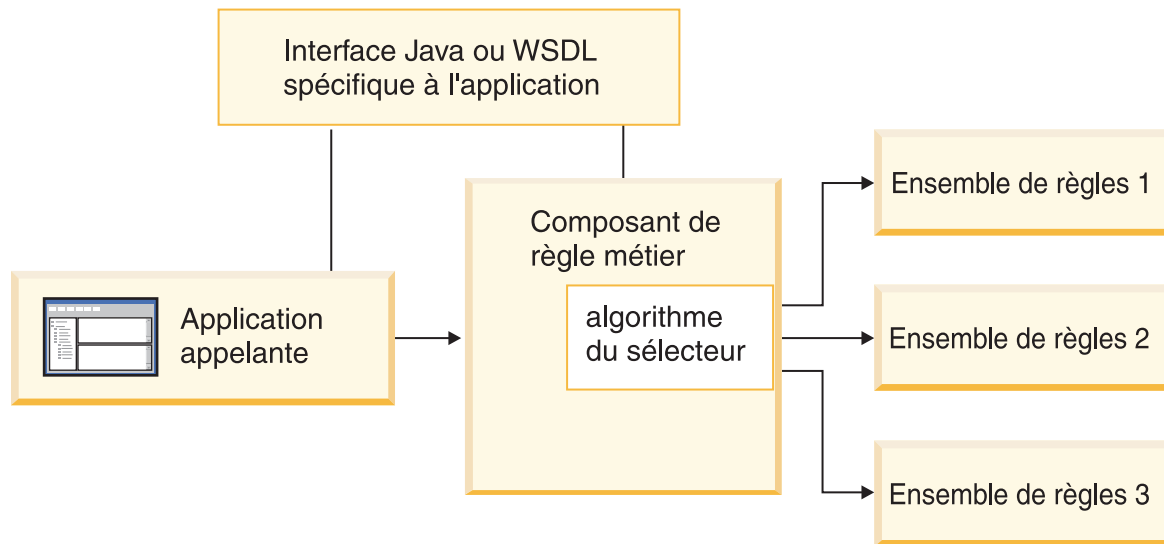


Les règles métier ne peuvent pas être utilisées dans un module de médiation. Elles ne peuvent être déployées que sur un serveur WebSphere Process Server.

## Sélecteurs

Les applications intégrées offrent de nombreuses possibilités d'interaction. Un *sélecteur* permet d'acheminer une opération d'une application client vers un des composants disponibles pour l'implémentation.

Le routage vers un composant est basé sur des dates. Voici un exemple de route basée sur une date : *deux semaines avant la rentrée des classes, accorder un prix spécial rentrée sur les articles scolaires.* Les entreprises peuvent utiliser de nombreuses routes de ce type, basées sur des dates. Un sélecteur décide de sélectionner une route plutôt qu'une autre lors de l'exécution en fonction d'une date. Par exemple, si une date se situe immédiatement avant la rentrée des classes, alors l'offre spéciale rentrée précédente est appelée. Cependant, si la date se situe vers la fin de l'année scolaire, une offre peut promouvoir les articles d'été.

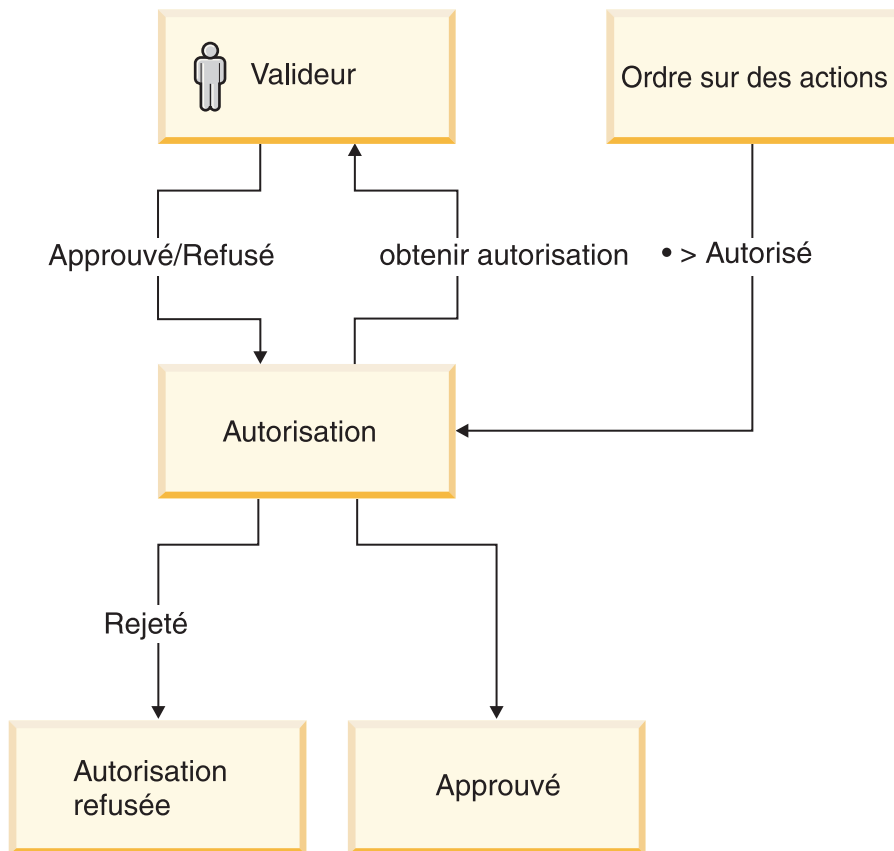


Un sélecteur ne peut pas être utilisé dans un module de médiation. Il ne peut être déployé que sur un serveur WebSphere Process Server.

## Tâche manuelle

Un composant de type *tâche manuelle* implémente une tâche effectuée par une personne. Il représente l'engagement d'une personne dans un processus métier.

Des personnes doivent parfois intervenir dans un processus métier. Par exemple, un client souhaite acheter un article dont le prix dépasse son plafond de crédit autorisé. Une tâche manuelle vous permet alors d'intervenir pour outrepasser une règle métier qui empêche ce client d'effectuer l'achat. Une tâche manuelle peut avoir des attributs, comme la définition du propriétaire de la tâche et la définition d'un processus d'escalade en cas d'indisponibilité de la personne désignée. Le composant de type tâche manuelle reconnaît que de nombreux processus requièrent l'intervention humaine pour des tâches, telles que la révision, la recherche et l'approbation.



Une tâche manuelle ne peut pas être utilisée dans un module de médiation. Elle ne peut être déployée que sur un serveur WebSphere Process Server.

## Mappe d'interface

Une *mappe d'interface* résout les différences entre les interfaces des composants qui interagissent.

Il existe fréquemment des différences entre les interfaces des composants devant interagir. Ces différences s'expliquent par le fait que dans WebSphere Integration Developer, vous assemblez souvent des composants qui ont été créés pour différentes applications. Un des avantages de WebSphere Integration Developer consiste à les réutiliser pour créer une nouvelle application, ce qui évite de recoder des composants similaires. Toutefois, des modifications doivent généralement être apportées.

Par exemple, deux composants peuvent comporter des méthodes qui exécutent fondamentalement la même opération, mais ont des noms différents tels que `getCredit` et `getCreditRating`. Ils peuvent également avoir des noms d'opérations différents et les opérations peuvent être associées à des types de paramètres distincts. Une mappe d'interface met en correspondance les opérations et les paramètres de ces méthodes de façon à ce que les différences soient résolues et que les deux composants puissent interagir. Une mappe d'interface est comparable à un pont entre les interfaces des deux composants leur permettant d'être connectées en dépit de leurs différences.

Une mappe d'interface existe indépendamment des composants qui l'utilisent, ce qui signifie qu'il n'est pas nécessaire de modifier les composants eux-mêmes.

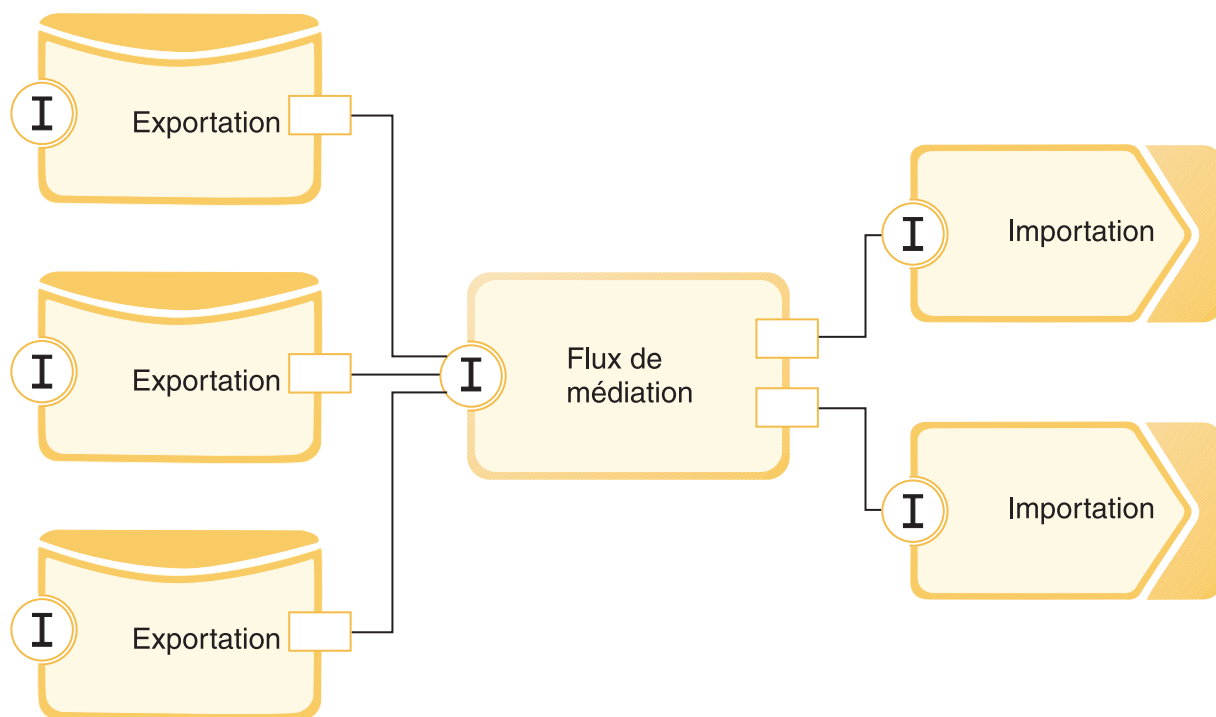
Une mappe d'interface ne peut pas être utilisée dans un module de médiation. Elle ne peut être déployée que sur un serveur WebSphere Process Server.

## Flux de médiations

La *médiation* est un moyen d'intervenir de façon dynamique entre des services. Un *flux de médiations* implémente une médiation.

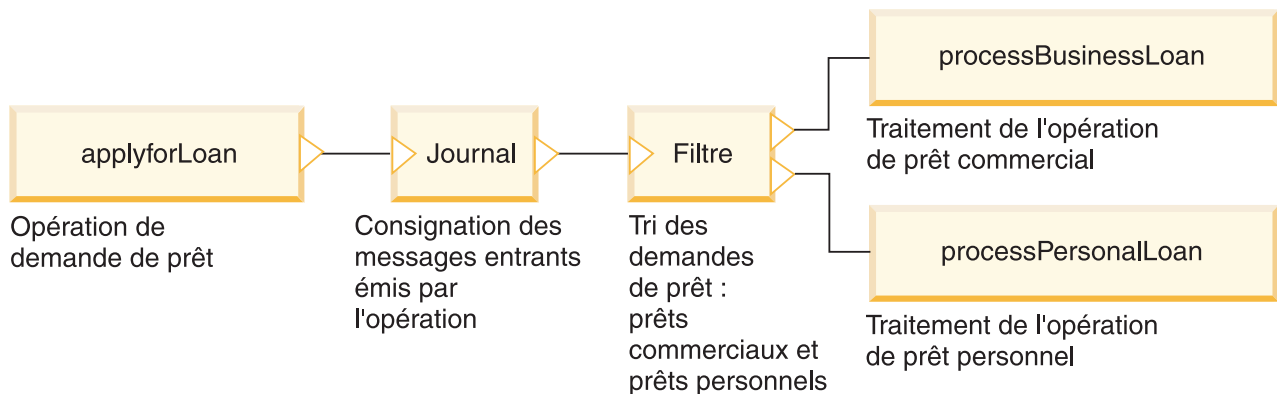
La médiation offre plusieurs fonctions utiles. Par exemple, vous pouvez l'utiliser pour convertir des données d'un service dans un format acceptable pour un service ultérieur. La journalisation vous permet de consigner des messages d'un service avant qu'ils soient envoyés au suivant. Le routage permet d'acheminer des données d'un service vers un service approprié, identifié par le flux de médiations. Une médiation fonctionne indépendamment des services auxquels elle se connecte. Elle apparaît dans l'éditeur d'assemblage comme un composant de flux de médiations entre les exportations et les importations.

Dans le diagramme suivant, trois demandeurs ou exportations de services envoient leurs données de sortie à l'interface du composant de flux de médiations. Ce composant achemine ensuite les données appropriées vers les fournisseurs ou importations de services.



Un flux de médiations est une construction de type flux créée avec l'éditeur de flux de médiations. La sélection d'un composant de flux de médiations dans l'éditeur d'assemblage lance l'éditeur de flux de médiations. Dans l'éditeur de flux de médiations, une opération d'un service, tel que l'exportation ou le demandeur de services, est mappée avec l'opération d'un autre service, tel que l'importation ou le fournisseur de services, conjointement avec les fonctions fournies par l'éditeur de flux de médiations. Ces fonctions sont appelées *primitives de médiation* et sont connectées dans un flux de médiations, comme illustré dans le diagramme ci-dessous. Les primitives de médiation sont fournies par IBM ; par ailleurs, vous pouvez créer vos propres primitives personnalisées. Les primitives de médiation peuvent s'appliquer au contenu des messages et à leur contexte, qui correspond aux informations spécifiques à la liaison, telles que les en-têtes SOAP ou JMS ou les propriétés définies par l'utilisateur.

Dans le diagramme suivant, une opération, `applyforLoan`, envoie d'abord un message à une primitive de journalisation, `Journal`, qui enregistre le message. La fonction `Journal` envoie le message à la primitive `Filtre` qui, selon le type du message, l'achemine vers une opération `processBusinessLoan` ou `processPersonalLoan`.



Comme indiqué dans la section Modules, un module de médiation est associé à un composant de flux de médiations. Il contient au maximum un composant de flux de médiations ainsi que zéro ou plusieurs composants Java étendant le composant de flux de médiations. Un module de médiation peut être déployé sur un serveur WebSphere Process Server ou WebSphere Enterprise Service Bus.

## Références autonomes

Les *références autonomes* renvoient à des applications non définies comme des composants SCA (par exemple, JavaServer Pages ou servlets). Elles permettent à ces applications d'interagir avec les composants SCA (Service Component Architecture).

Les références autonomes ne sont associées à aucune interface ni aucune implémentation (étant donné que l'implémentation se situe hors de la portée du module). Un module peut ne pas contenir de référence autonome ou inclure un artefact de références autonomes. Les références autonomes sont pratiques dans la mesure où elles vous permettent d'utiliser les applications existantes avec les composants SCA (Service Component Architecture) créés dans WebSphere Integration Developer.

Les références autonomes peuvent être utilisées dans un module de médiation. Elles peuvent être déployées sur un serveur WebSphere Process Server ou WebSphere Enterprise Service Bus.

---

## Informations connexes

Plusieurs rubriques liées à l'architecture sont traitées dans cette section.

Ces rubriques fournissent des informations complémentaires concernant l'architecture du produit.

- «Développement de services à utiliser avec les services .NET»
- «Support bidirectionnel», à la page 25

## Développement de services à utiliser avec les services .NET

Si vous avez l'intention de développer des services à utiliser avec les services .NET, vous devez prendre en compte quelques remarques spéciales. Deux aspects sont à prendre en considération : transfert dans l'environnement de développement WebSphere Integration Developer de fichiers WSDL développés dans un environnement de développement .NET et exportation de ces fichiers depuis WebSphere Integration Developer pour qu'ils puissent être utilisés avec les services .NET. La principale différence entre les deux environnements de développement réside dans le fait que l'environnement .NET utilise des schémas en ligne, ce qui n'est pas le cas de WebSphere Integration Developer. Les schémas en ligne permettent d'inclure le schéma dans un fichier WSDL au lieu de spécifier son importation en tant que fichier séparé. Des informations sont disponibles dans WebSphere Integration Developer pour vous aider dans ces deux cas de figure.



La rubrique Schémas en ligne de la section Création d'interfaces indique comment importer des fichiers WSDL contenant des schémas en ligne tels que ceux contenus dans les fichiers WSDL .NET.

La rubrique Création de proxy à utiliser avec les services .NET de la section Assemblage du service métier montre comment publier vos services en externe pour qu'ils soient exploitables avec les services .NET. Cette rubrique traite tout particulièrement de la création de proxy devant fonctionner avec les services .NET.

## **Support bidirectionnel**

WebSphere Integration Developer fonctionne dans un environnement multilingue. Cela signifie qu'il peut afficher et manipuler des données représentées dans différentes langues. Plusieurs langues utilisant des scripts bidirectionnels (par exemple, l'arabe ou l'hébreu) sont incluses dans ce support. Ces langues s'écrivent de droite à gauche, tandis que les nombres et les segments de texte imbriqués en Latin (cyrillique, grec, etc.) sont intégrés de gauche à droite.

La section Présentation du support des scripts bidirectionnels dans IBM WebSphere Integration Developer traite du support des langues bidirectionnelles, et notamment de la configuration requise, de certains points techniques spécifiques à prendre en compte lors de l'utilisation du support et des restrictions applicables.



---

## Chapitre 3. Apprentissage des outils

WebSphere Integration Developer utilise des outils visuels pour créer des applications intégrées basées sur une architecture orientée service.

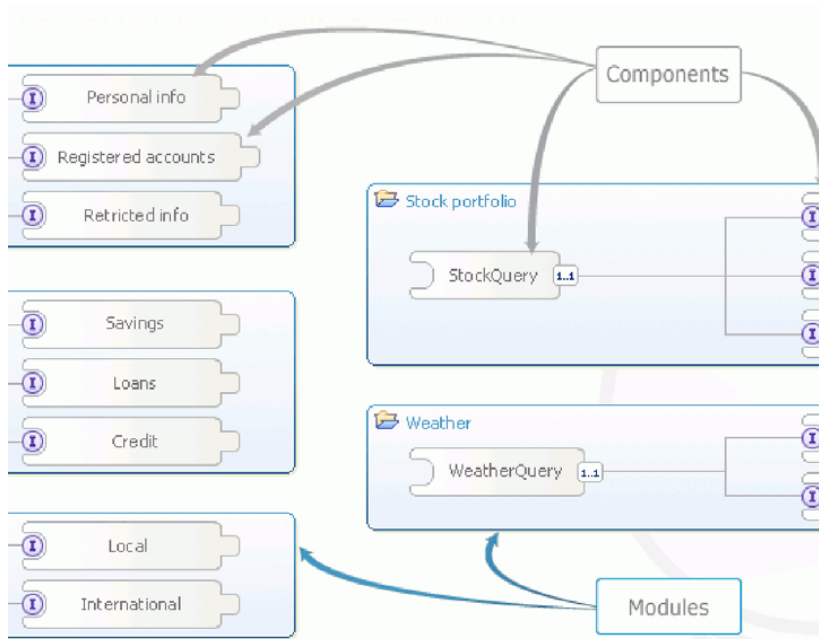
La vue d'ensemble de la vue Bienvenue propose l'aperçu de plus haut niveau sur les outils WebSphere Integration Developer, c'est-à-dire la présentation la plus abstraite. Les graphiques expliquent les fonctions des outils. Lisez la vue d'ensemble si vous débutez dans l'utilisation du produit et que vous voulez rapidement comprendre les tâches exécutables par ces outils. La section des tutoriels, qui se trouve également dans la vue Bienvenue, permet de visualiser les outils en cours d'exécution. La section des exemples permet une mise en route à sécurité intégrée. Enfin, lorsque vous commencez à développer des applications avec WebSphere Integration Developer, utilisez le centre de documentation qui propose, pour chaque outil, des informations détaillées, des informations conceptuelles, les tâches possibles ainsi que des informations de référence.

---

### Vue d'ensemble de la vue Bienvenue

La vue d'ensemble de la vue Bienvenue constitue l'aperçu de plus haut niveau sur les outils et les fonctions du produit. Dans cette vue d'ensemble, des graphiques illustrent les fonctions proposées par les principaux outils du produit.

La section Vue d'ensemble de la vue Bienvenue permet de se familiariser rapidement avec le produit avant de s'exercer à l'utiliser. La plupart des nouveaux utilisateurs doivent l'utiliser comme point de départ. Vous pouvez lancer un graphique sur un outil ou une rubrique en sélectionnant une icône dans la vue d'ensemble. Par exemple, les relations des composants et des modules sont présentées dans le diagramme ci-après. Vous pouvez également lancer une visite audiovisuelle du produit.

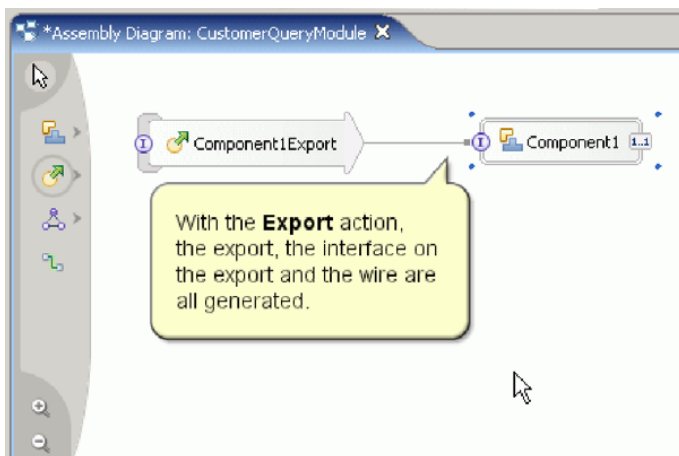


---

## Tutoriels de la vue Bienvenue

Les tutoriels vous permettent de visualiser les outils en action dans un mode à sécurité intégrée. Vous sélectionnez un tutoriel, puis vous l'observez exécuter certaines tâches comme s'il s'agissait d'un utilisateur.

Lorsque vous avez compris les concepts du produit, vous pouvez lancer un tutoriel pour vous familiariser avec les outils. Chaque tutoriel correspond à une démonstration vidéo d'un outil déterminé. Dans le tutoriel, des bulles d'aide décrivent les éléments visualisés. Les tutoriels WebSphere Integration Developer se trouvent dans la section sur l'apprentissage par observation, dans la galerie des tutoriels de la vue Bienvenue.



---

## Exemples de la vue Bienvenue

Ces exemples proposent une présentation pratique de l'utilisation de WebSphere Integration Developer pour développer des solutions de services métier dans un environnement à sécurité intégrée.

Ils se trouvent dans la galerie d'exemples de la vue Bienvenue. La sélection d'un exemple entraîne plusieurs options. Vous pouvez créer l'application vous-même en suivant les instructions pas à pas ou choisir la création automatique.

Les exemples ont une portée variable. Un exemple de technologie se concentre sur un outil particulier exécutant une tâche spécifique. Un exemple d'application utilise plusieurs outils pour produire un résultat plus complexe. Un scénario est un exemple long dans lequel une grande application est créée à l'aide de nombreux outils. Chaque exemple indique le temps requis pour la création.

## Business state machine (simple)

This sample demonstrates how a business state machine can be used to moderate a sales order transaction. Specifically, the state machine emulates an on-line brokerage that manages the selling of a share.

The following tools are used in this application:

- Business state machine editor
- Human task editor

To import the ready-made sample, click the **Import** link below and click **Finish** in the opened wizard. See **Running instructions** to run the imported code.

or

If you want to build the sample for yourself, click **Step-by-step instructions**.

### Ready-made sample

[Import](#) [Running instructions](#)

### Build it yourself

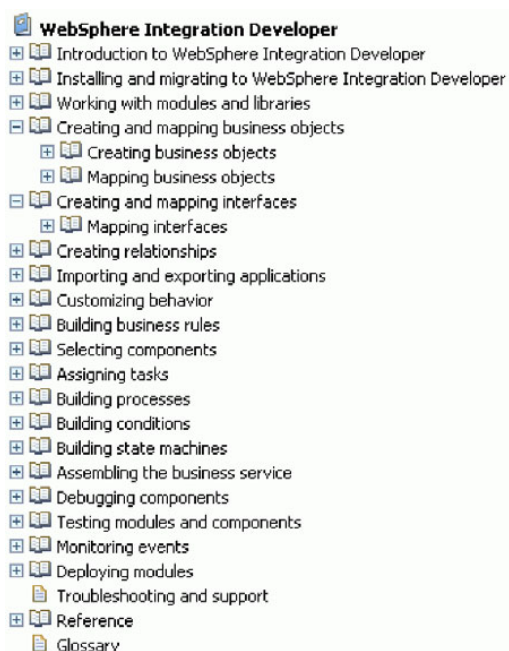
[Step-by-step instructions](#)

---

## Centre de documentation

Le centre de documentation fournit des informations complètes sur le produit. Outre des tutoriels et des exemples, il propose, pour chaque outil WebSphere Integration Developer, des informations supplémentaires sur les concepts, les tâches et les éléments de référence.

La fonction de navigation offerte par le centre de documentation permet d'accéder rapidement à l'information voulue. Vous pouvez également effectuer une recherche dans tout le centre de documentation. Dans les résultats de la recherche, indiquez votre choix pour consulter les différentes rubriques qui contiennent des informations approfondies sur le sujet ainsi que des liens vers des rubriques connexes.





---

## Remarques

L'utilisation de la documentation XDoclet incluse dans ce produit IBM a été autorisée sous la mention de copyright suivante : Copyright (c) 2000-2004, XDoclet Team. All rights reserved.

Parties basées sur des *modèles de conception : éléments de logiciel orienté-objet réutilisables*, par Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides, Copyright (c) 1995 by Addison-Wesley Publishing Company, Inc. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de cette documentation 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 EMEA Director of Licensing IBM Europe Middle-East Africa  
Tour Descartes  
92066 Paris-La Défense Cedex 50  
France  
Pour le Canada, veuillez adresser votre courrier :  
IBM Director of Commercial Relations  
IBM Canada Ltd  
3600 Steeles Avenue East Markham,  
Ontario  
L3R 9Z7 Canada

Les informations sur les licences concernant les produits IBM utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales : LES INFORMATIONS SONT LIVREES EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAULT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document 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 :

*Intellectual Property Dept. for Rational  
Software  
IBM Corporation  
20 Maguire Road  
Lexington, Massachusetts 02421-3112  
U.S.A.*

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

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

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

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

#### LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque



forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation IBM.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

(C) (nom de votre société) (année). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. (C) Copyright IBM Corp. 2000, 2005. All rights reserved.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

## **Documentation sur l'interface de programmation**

La documentation sur l'interface de programmation aide les utilisateurs à créer des applications en utilisant le produit.

Les interfaces de programmation génériques vous permettent d'écrire des applications, qui bénéficient des services proposés par les outils du produit.

Toutefois, lesdites informations peuvent également contenir des données de diagnostic, de modification et d'optimisation qui permettent de déboguer votre application.

**Avertissement :** N'utilisez pas les informations de diagnostic, de modification et d'optimisation en guise d'interface de programmation car elles peuvent être modifiées sans préavis.

## **Marques**

Voir <http://www.ibm.com/legal/copytrade.shtml>.







SC11-2367-01

