



## **Technische Produktübersicht**





## Technische Produktübersicht

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen in "Bemerkungen" am Ende dieses Buchs gelesen werden.

---

# Inhaltsverzeichnis

<b>Kapitel 1. Integration zwischen Geschäftsbereichen . . . . .</b>	<b>1</b>	Java-Objekte . . . . .	25
<b>Kapitel 2. Integration zwischen Unternehmen . . . . .</b>	<b>3</b>	BPEL-Prozess. . . . .	25
<b>Kapitel 3. WebSphere Integration Developer . . . . .</b>	<b>5</b>	Statusmaschinen. . . . .	26
Standards . . . . .	6	Geschäftsregeln . . . . .	27
Rolle des Integrationsentwicklers . . . . .	6	Selektoren . . . . .	28
<b>Kapitel 4. Servicekomponenten. . . . .</b>	<b>9</b>	Human Task . . . . .	29
<b>Kapitel 5. Servicedatenobjekte (SDO) . . . . .</b>	<b>13</b>	Schnittstellenzuordnung . . . . .	30
<b>Kapitel 6. Servicequalifikationsmerkmale . . . . .</b>	<b>15</b>	Mediationsablauf . . . . .	31
<b>Kapitel 7. Module . . . . .</b>	<b>17</b>	Eigenständige Verweise . . . . .	32
<b>Kapitel 8. Importe und Exporte . . . . .</b>	<b>21</b>	<b>Kapitel 10. Zugehörige Informationen . . . . .</b>	<b>33</b>
Export- und Importbindingtypen für Services . . . . .	21	<b>Kapitel 11. Sicht 'Willkommen' - Übersicht. . . . .</b>	<b>35</b>
Geeignete Bindings auswählen . . . . .	22	<b>Kapitel 12. Lernprogramme in der Sicht 'Willkommen' . . . . .</b>	<b>37</b>
<b>Kapitel 9. Serviceimplementierungstypen . . . . .</b>	<b>25</b>	<b>Kapitel 13. Beispiele in der Sicht 'Willkommen' . . . . .</b>	<b>39</b>
		<b>Kapitel 14. Informationszentrum . . . . .</b>	<b>41</b>
		<b>Bemerkungen. . . . .</b>	<b>43</b>



---

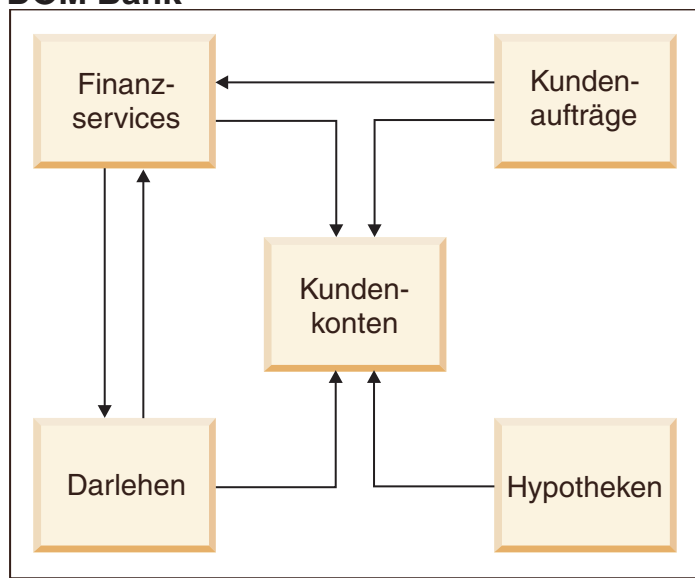
## Kapitel 1. Integration zwischen Geschäftsbereichen

Geschäftsbereiche arbeiten heutzutage immer mehr zusammen, so dass eine enge Integration zwischen ihren Anwendungen erforderlich wird.

Ehemals autonome Geschäftsbereiche werden nun integriert, weil die Technologie die Verbindung ermöglicht und weil es aus Gründen der Effizienz nahe liegt, dass sie kooperativ zusammenarbeiten, um die Gemeinkosten zu senken und ein maximales Ergebnis zu erzielen. Zudem werden Geschäftsbereiche durch ein gemeinsames Unternehmensziel vereint. Ein Marketingbereich und ein Forschungs- und Entwicklungsbereich wollen beide ein gewinnbringendes Produkt auf den Markt bringen. Die Integration des Marktknow-hows in die Informationen zur Produktentwicklung steigen die Chancen, ein erfolgreiches Produkt zu kreieren. Die Zusammenarbeit zwischen Geschäftsbereichen ermöglicht Unternehmen zudem, ihre vorhandenen Geschäftsanwendungen durch Zulassung ihrer Wiederverwendung in verschiedenen Geschäftskontexten vorteilhaft zu nutzen.

Die Integration zwischen Geschäftsbereichen gestaltet sich einfacher als die Integration zwischen Unternehmen, da das Sicherheitsrisiko geringer ist und die Verwaltung der Interaktionen zwischen den Bereichen nicht so schwierig sein sollte. Die Geschäftsbereiche arbeiten wahrscheinlich mit den gleichen Protokollen, Betriebssystemen und Maschinensprachen. Es handelt sich eher um eine relativ homogene Umgebung. Der Schlüssel indes besteht darin, die richtigen Tools zur Hand zu haben, um die Integration der Anwendungen rasch vollziehen zu können. Im folgenden Diagramm hat das Bankhaus DOM verschiedene Geschäftsbereiche, die gegenseitig die Informationen der anderen nutzen müssen. Noch vor einigen Jahren konnte das Unternehmen DOM bestehen, indem es Kopien der Informationen aus einem Geschäftsbereich in gedruckter Form einem anderen zukommen ließ, wozu eigene Systeme und Anwendungen eingerichtet waren. Heute muss die Bank DOM integrierte Anwendungen entwickeln, die ihre Geschäftsbereiche umfassen, wenn sie mit der Konkurrenz mithalten will.

### DOM Bank







---

## Kapitel 2. Integration zwischen Unternehmen

Die Triebkräfte der Integration von Anwendungen zwischen Geschäftsbereichen wirken auch zwischen Unternehmen, wenn Partnerschaften oder Übernahmen gemeinsame Daten und Prozesse erforderlich machen.

Technologie ermöglicht die Verknüpfung zwischen Unternehmen in beiderseits vorteilhaften Bereichen. Ein Automobilhersteller kann zum Beispiel einen integrierten Prozess mit einem Reifenzulieferer einrichten, so dass der Zulieferer automatisch benachrichtigt wird, wenn der Lagerbestand an Reifen gering ist. Die Integration zwischen Unternehmen wird von wirtschaftlichen Notwendigkeiten vorangetrieben. Engere Verbindungen zwischen Unternehmen bedeuten weniger Verzögerungen und weniger Aufwand. Solche automatisierten Prozesse sorgen dafür, dass Mitarbeiter weniger Zeit auf die Bearbeitung von Geschäftsvorgängen zwischen Unternehmen aufwenden und Reisekosten und Zeitaufwände für persönliche Besprechungen erheblich verringert werden können. Verwaltungskosten werden ebenso verringert, Verarbeitungszeiten zwischen Benachrichtigung, Lieferung und Rechnungsstellung verkürzt.

Verschiedene Unternehmen haben jedoch verschiedene Geschichten. Ihre Anwendungen sind in unterschiedlichen Sprachen auf verschiedenen Plattformen mit unterschiedlichen Kommunikationsprotokollen codiert. Darüber hinaus bestehen bei der Arbeit mit verschiedenen Unternehmen höhere Sicherheitsrisiken. Ungeachtet der Vorteile oder sogar der Notwendigkeit der Integration zwischen Unternehmen kann der Aufwand an Entwicklungszeit ohne die richtigen Werkzeuge beträchtlich sein.

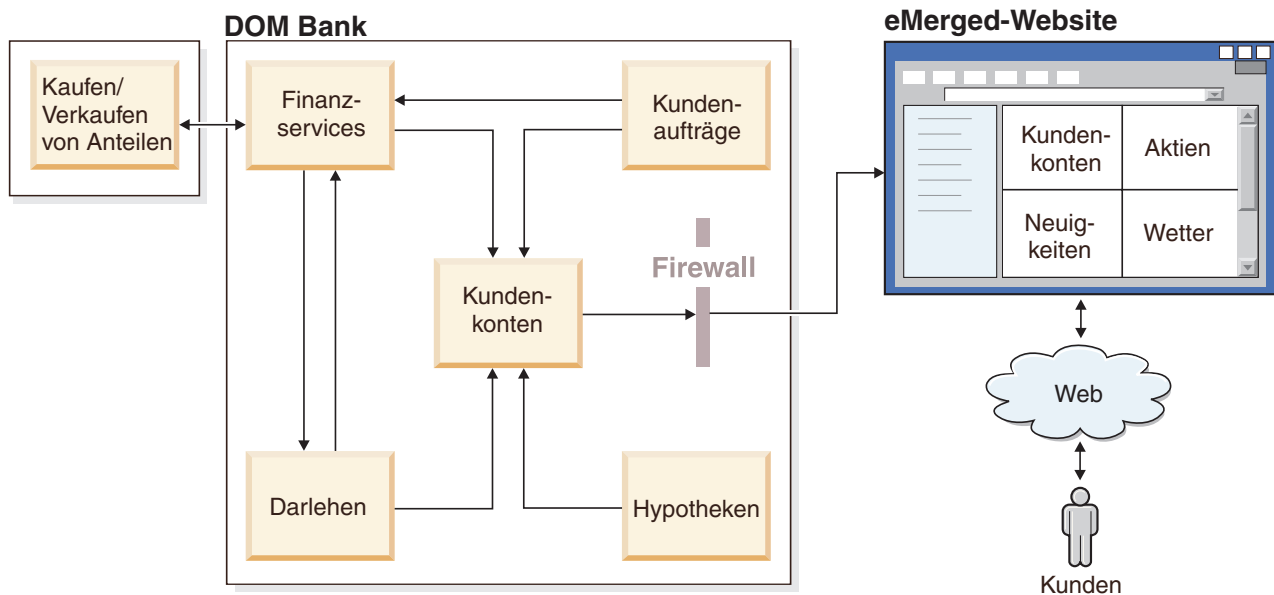


## Kapitel 3. WebSphere Integration Developer

WebSphere Integration Developer ist die Antwort auf die Herausforderungen im Bereich Geschäftsintegration, mit denen Unternehmen jeden Tag konfrontiert werden. Das Produkt wurde als vollständige Integrationsentwicklungsumgebung für Personen konzipiert, die integrierte Anwendungen erstellen. Zur Vereinfachung und Beschleunigung der Entwicklung integrierter Anwendungen bietet diese Umgebung eine Abstraktionsebene, die graphisch dargebotene Komponenten, mit denen Sie arbeiten, von der zugrunde liegenden Implementierung trennt.

Integrierte Anwendungen sind nicht banal. Sie können Anwendungen auf Enterprise Information Systems (EIS-Systemen) aufrufen, Geschäftsprozesse abteilungs- oder unternehmensübergreifend mit einbeziehen und Anwendungen lokal oder remote aufrufen, die in einer ganzen Reihe verschiedener Sprachen geschrieben sind und auf unterschiedlichen Betriebssystemen ausgeführt werden. Die eMerged Corporation entstand zum Beispiel aus der Fusion der DOM Bank mit M&M Discount Brokers. Der Unternehmenszusammenschluss beinhaltete alle oben erwähnten Aspekte: Anwendungen auf EIS-Systemen, Geschäftsprozesse und in den jeweiligen Unternehmen vorhandene Anwendungen mussten zwischen den Unternehmen gemeinsam nutzbar gemacht und einer neuen Kundengruppe in einer nahtlosen Weise zur Verfügung gestellt werden. Das Unternehmen eMerged bewältigte diese Aufgabe jedoch wie im folgenden Diagramm gezeigt. Kunden beider früheren getrennten Unternehmen können auf all ihre Finanzinformationen online zugreifen.

### M&M Discount Brokers



Das Unternehmen eMerged nutzte die Tools von WebSphere Integration Developer, um die integrierten Anwendungen für sich und ihre Kunden aufzubauen. Diese Tools stellen Anwendungen, einschließlich Anwendungen, die sich remote auf EIS-Systemen befinden, sowie Geschäftsprozesse als Komponenten dar. Die Komponenten werden mit Hilfe graphisch orientierter Editoren erstellt und zu anderen integrierten (d. h. aus einer Gruppe von Komponenten erstellte) Anwendungen zusammengesetzt. Die graphisch orientierten Editoren bilden eine Abstraktionsschicht zwischen den Komponenten und ihren Implementierungen. Ein Entwickler, der mit den Tools arbeitet, kann eine integrierte Anwendung erstellen, ohne über detaillierte Kenntnisse über die zugrunde liegende Implementierung der einzelnen Komponenten verfügen zu müssen.

Die Tools ermöglichen sowohl ein Top-down-Designstrategie zum Aufbau einer integrierten Anwendung, bei der die Implementierung für eine oder mehrere Komponenten nicht vorhanden ist und später hinzugefügt wird, als auch einen Bottom-up-Ansatz, bei dem die Komponenten bereits implementiert sind und der Entwickler sie per Drag-and-Drop in einem graphisch orientierten Editor zusammenfügt und anschließend durch Einfügen von Verbindungslinien einen logischen Ablauf zwischen ihnen festlegt. Eine Debugging- und Testumgebung ermöglicht umfassendes Testen, bevor Ihre Anwendungen auf einem Produktionsserver zum Einsatz gebracht werden. Durch Einstellung von Überwachungspunkten haben Sie die Möglichkeit, die Verwendung der Anwendung in Echtzeit zu beobachten, um eine Feinabstimmung auf optimale Leistung vorzunehmen.

Die Tools von WebSphere Integration Developer basieren auf einer serviceorientierten Architektur. Komponenten sind Services, und eine integrierte Anwendung mit vielen Komponenten ist ein Service. Die erstellten Services entsprechen den branchenweit führenden Standards. Geschäftsprozesse, die ebenfalls zu Komponenten werden, werden auf ähnliche Weise mit benutzerfreundlichen graphisch orientierten Tools erstellt, die der standardisierten Sprache Business Process Execution Language (BPEL) entsprechen. WebSphere Integration Developer ist sowohl für Windows- als auch für Linux-Plattformen verfügbar.

Die Tools von WebSphere Integration Developer bieten unter anderem folgende Vorteile:

- Sie sind leicht erlernbar.
- Sie können auf komplexe Integrationssituationen angewendet werden.
- Sie ermöglichen eine rasche Erstellung von Anwendungen, die branchenweiten Standards entsprechen.

---

## Standards

Anwendungen, die durch WebSphere Integration Developer erstellt werden, entsprechen branchenweiten Standards für die serviceorientierte Architektur.

Niemand möchte Anwendungen erstellen, die an proprietären Code gebunden sind, der innerhalb weniger Jahre vielleicht schon nicht mehr unterstützt wird oder kostspielige Lizenzierungsgebühren erfordert. Die auf Standards basierende Integration ist daher ein wesentlicher Aspekt von WebSphere Integration Developer. Für die Konnektivität werden J2EE Connector Architecture-Standards genutzt. Für asynchrone Nachrichtenübermittlung, die häufig in großen Anwendungen mit obligatorischer garantierter Datenzustellung zum Einsatz kommt, wird der Java Message Service-Standard (JMS-Standard) verwendet. WebSphere Integration Developer kann Web-Services, die auf dem Simple Object Access Protocol (SOAP) basieren, problemlos integrieren. Zur Beschreibung eines Service wird der anerkannte Standard Web Services Description Language (WSDL) verwendet. Zur Definition eines Geschäftsprozesses wird der Standard Business Process Execution Language (BPEL) verwendet.

Diese auf Standards basierenden Schnittstellen und Komponenten bilden eine endeoffene und modular aufgebaute Architektur. Proprietäre Elemente können jedoch nicht ausgeschlossen werden. Der Zugriff auf solche Elemente erfolgt unter Verwendung standardisierter Schnittstellen. Dies bedeutet, dass in WebSphere Integration Developer erstellte Anwendungen zum Beispiel mit .NET-Anwendungen interagieren können. Der Abschnitt über die Architektur enthält einen Link zur vollständigen Service Component Architecture, über den auf eine umfassende Liste der zahlreichen unterstützten Standards zugegriffen werden kann.

---

## Rolle des Integrationsentwicklers

Der primäre Benutzer von WebSphere Integration Developer ist der Integrationsentwickler. Diese Person kann durch Verwendung grafisch orientierter Tools eine komplexe integrierte Anwendung erstellen, ohne weitreichende Kenntnisse über die zugrunde liegende Implementierung zu benötigen.

WebSphere Integration Developer stellt Anwendungen und Geschäftsprozesse als Komponenten dar. Die Implementierung der Komponenten bleibt verborgen, die Interaktion der Komponenten erfolgt über Schnittstellen. Daher benötigen Integrationsentwickler keine tiefen Kenntnisse über die zugrunde liegende

Implementierung der Komponenten, um eine integrierte Anwendung zu erstellen, die mit solchen Komponenten arbeitet. Integrationsentwickler verfügen indessen meist über ein breit gefächertes technisches Verständnis auf dem Gebiet der Integration, da sie einige Erfahrung mit EIS-Systemen und Geschäftsprozessen sowie mit in Java oder anderen Sprachen codierten Anwendungen mitbringen müssen. Zum Beispiel verfügt ein Architekt über ein allgemeines Wissen darüber, wie ein System funktioniert, ohne die Funktionsweise der einzelnen Komponenten genau zu kennen. Ebenso wie der Architekt kann der Integrationsentwickler diejenige Person in einem Unternehmen sein, die die Gesamtanwendung entwirft. Die Implementierung der einzelnen Komponenten wird anschließend von anderen Mitarbeitern codiert.

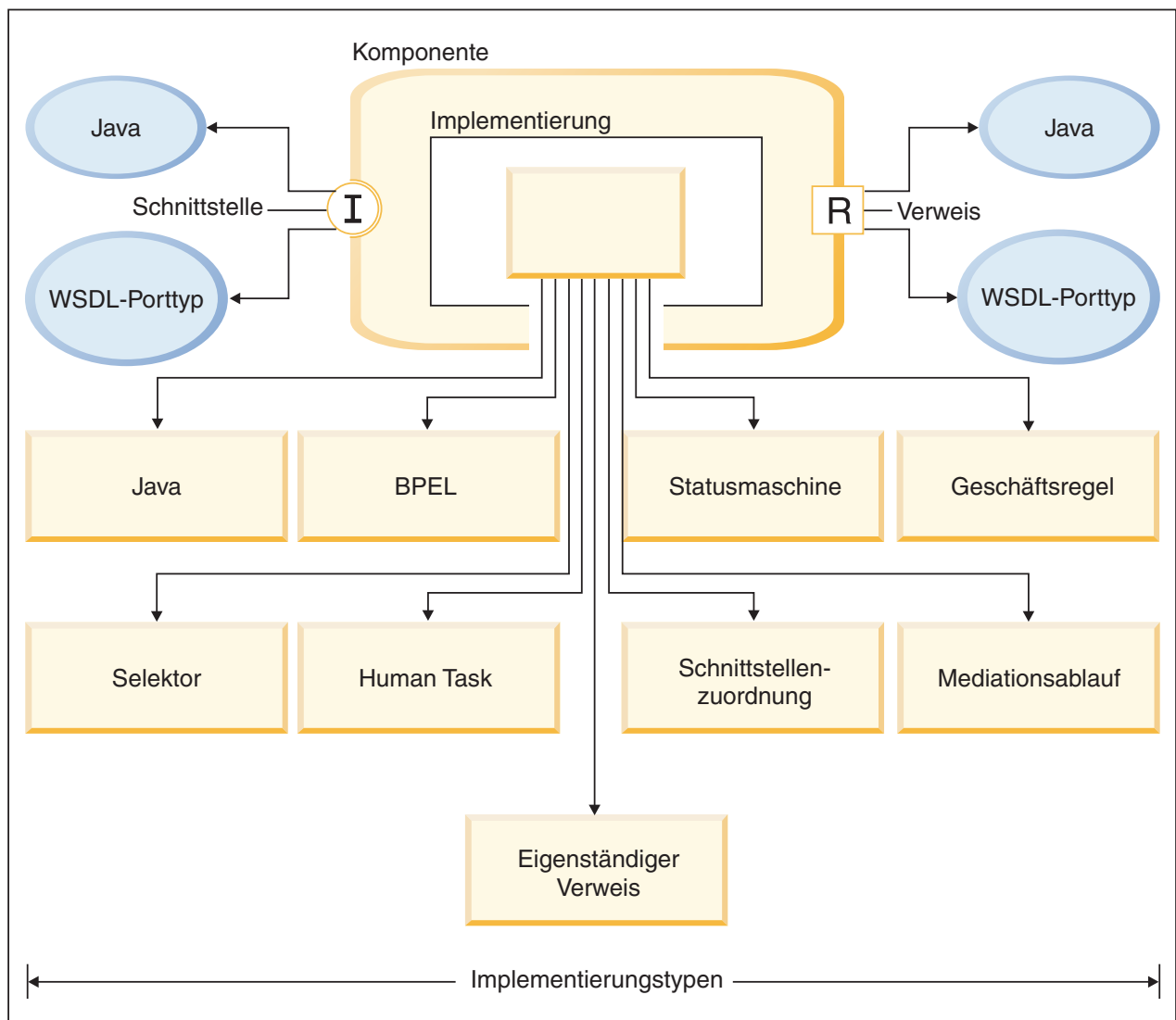


## Kapitel 4. Servicekomponenten

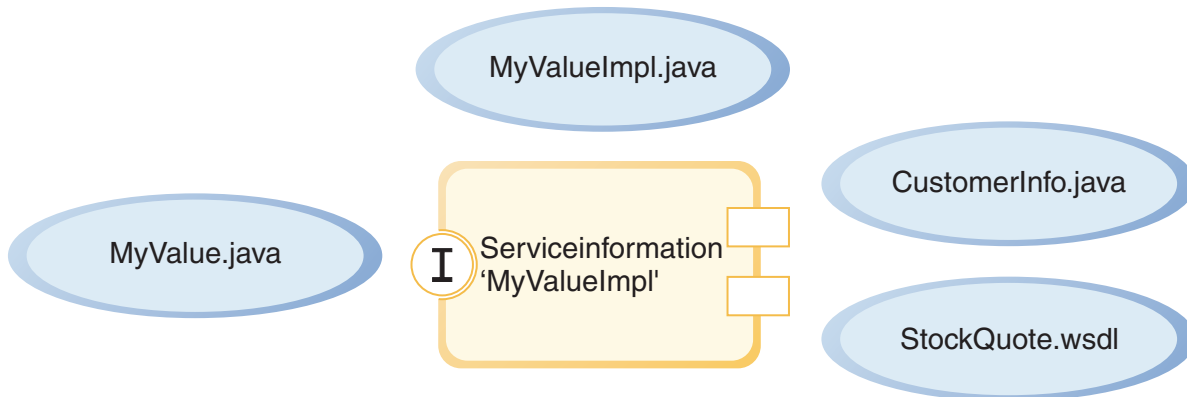
Eine Servicekomponente konfiguriert eine Serviceimplementierung. Eine Servicekomponente wird in einem Standardblockdiagramm dargestellt.

Eine Komponente besteht aus einer Implementierung, die bei Verwendung der Tools von WebSphere Integration Developer verborgen bleibt, mindestens einer Schnittstelle, die ihre Eingaben, Ausgaben und Fehler definiert, und keinem oder mehreren Verweis(en). Ein Verweis gibt die Schnittstelle eines anderen Service oder einer anderen Komponente an, die diese Komponente benötigt oder verwendet. Eine Schnittstelle kann in einer von zwei Sprachen definiert sein: als WSDL-Porttyp oder inJava. Eine Schnittstelle unterstützt synchrone und asynchrone Interaktionsstile. Die Implementierung einer Komponente kann in verschiedenen Sprachen vorliegen.

Ein Schnittstellentyp kann in WSDL oder in Java vorliegen. Wenn jedoch mehrere Schnittstellen vorhanden sind, können Sie WSDL nicht mit Java kombinieren.



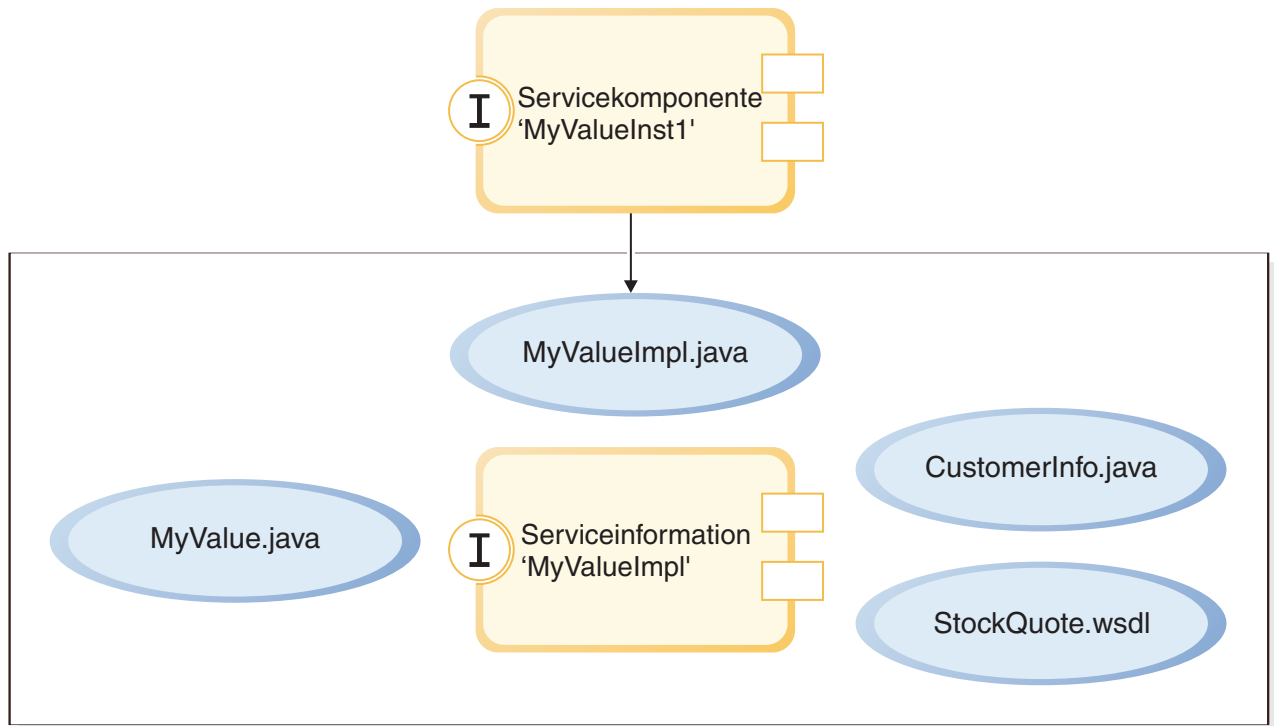
In der folgenden Abbildung befindet sich eine Komponente in der Mitte. Ihre Implementierung 'MyValueImpl' ist wie die zugehörige Schnittstelle in Java vorhanden. Sie verfügt über zwei Verweise: eine weitere Java-Schnittstelle und eine WSDL-Schnittstelle.



Bei der Arbeit mit einer solchen Komponente wird Ihnen effektiv nur die Komponente selbst angezeigt, wie in der folgenden Abbildung zu sehen ist. Ein Verweis auf diese Komponente von einer anderen Komponente würde graphisch durch eine Linie zur entsprechenden Schnittstelle dargestellt werden. Ein Verweis von dieser Komponente würde durch eine Linie von ihrem Referenzpunkt zur Schnittstelle einer anderen Komponente dargestellt werden. Ein Verweis stellt einen Service dar, der diese Komponente verwendet. Da ein Verweis und nur die Schnittstelle angegeben wird, kann der Autor der Komponentenimplementierung das Binding dieses Verweises an einen tatsächlichen Service bis zu einem späteren Zeitpunkt verzögern. Zu einem späteren Zeitpunkt holt der Integrationsspezialist dies nach, indem er eine Verbindung von dem Verweis zu der Schnittstelle einer anderen Komponente oder eines anderen Imports herstellt. Diese lose Verbindung, die eine Verzögerung des Bindings und eine erneute Verwendung der Implementierungen ermöglicht, ist einer der Hauptgründe für die Verwendung der Servicekomponentenarchitektur von WebSphere Integration Developer.

Eine Komponente kann auch Eigenschaften und Qualifikationsmerkmale besitzen. Ein Qualifikationsmerkmal ist eine Servicequalitätsanweisung (QoS - Quality of Service) für Schnittstellen und Verweise für die Ausführung.







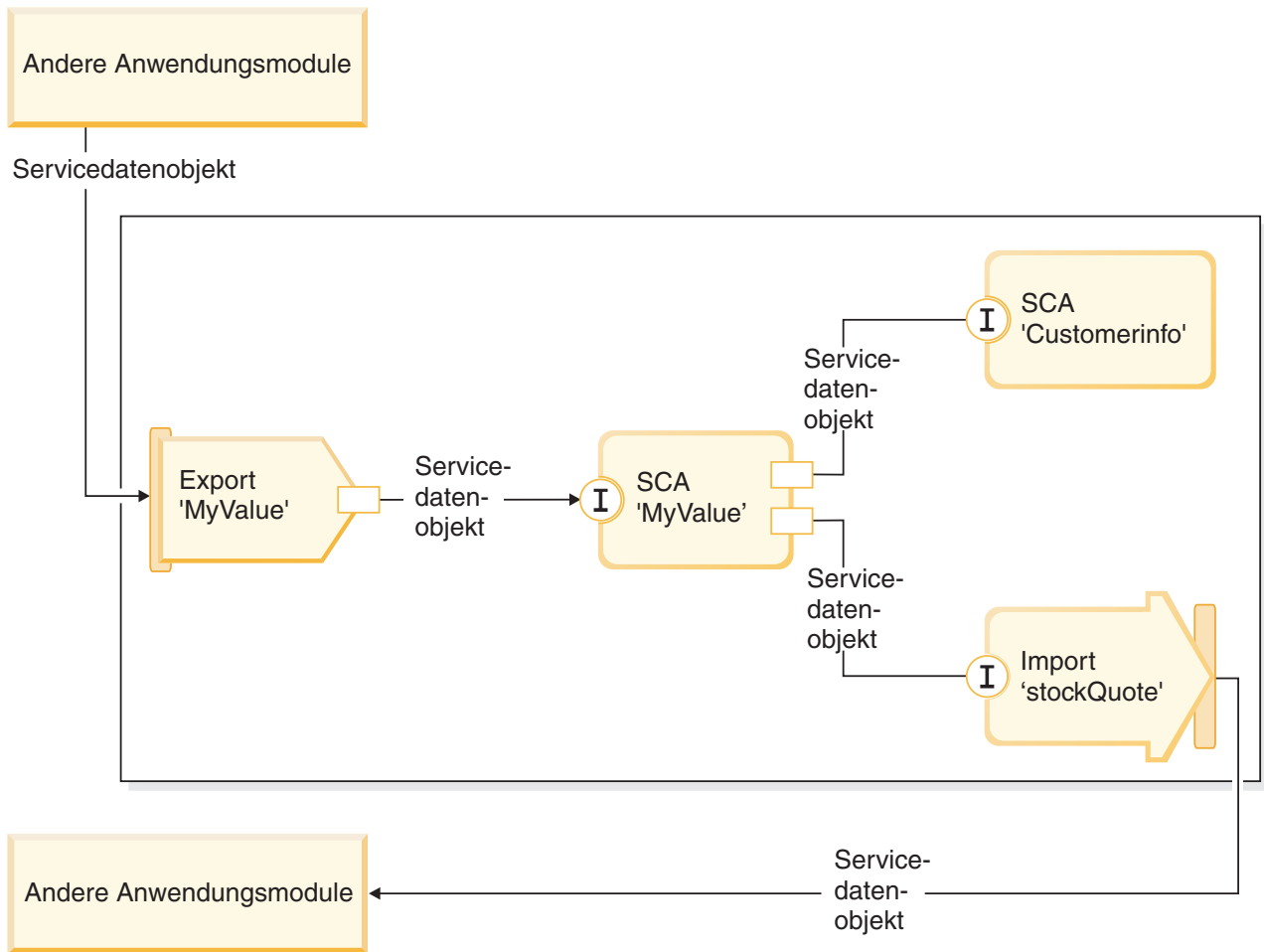
---

## Kapitel 5. Servicedatenobjekte (SDO)

Servicedatenobjekte (SDO) ergänzen die Komponente 'Service Component Architecture'. Die Service Component Architecture definiert die Services als Komponenten sowie die Konnektivität zwischen diesen Komponenten. Servicedatenobjekte definieren die Daten, die zwischen den Komponenten fließen.

Jede Komponente übergibt Informationen als Eingabe und Ausgabe. Wenn ein Service aufgerufen wird, werden Datenobjekte bei Verwendung eines WSDL-Porttyps als XML-Dokument mit Dokumentliteralcodierung und bei Verwendung einer Java-Schnittstelle als Java-Objekt übergeben. Datenobjekte sind die bevorzugte Form für Daten und Metadaten in Service Component Architecture-Services. Ähnlich wie Komponenten trennen Servicedatenobjekte das Datenobjekt von seiner Implementierung. Zum Beispiel könnte eine Komponente mit Bestellungen interagieren, während die Bestellung selbst mit JDBC, EJB usw. arbeitet, um die Aktualisierungen an den Daten vorzunehmen. Servicedatenobjekte ermöglichen es dem Integrationsentwickler, sich auf die Arbeit mit Geschäftsartefakten zu konzentrieren. Servicedatenobjekte sind für den Integrationsentwickler transparent. Sie werden von Servicedatenobjekte Java Specification Request (JSR) definiert.

Im folgenden Diagramm werden Servicedatenobjekte von einem externen Service an einen Export, von einem Export an eine Komponente, von einer Komponente an eine Komponente, von einer Komponente an einen Import und von einem Import an einen Service übergeben. Importe und Exporte werden im später folgenden Abschnitt Importe und Exporte besprochen.



---

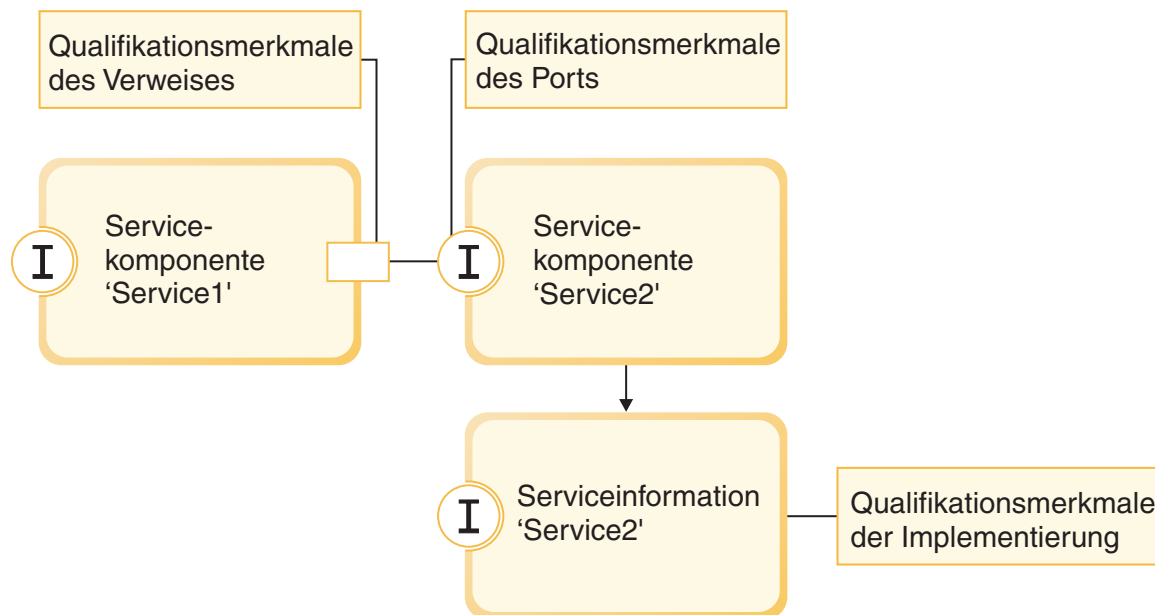
## Kapitel 6. Servicequalifikationsmerkmale

Eine Anwendung teilt ihre Servicequalitätsanforderungen (Quality of Service - QoS) der Laufzeitumgebung durch die Angabe von Servicequalifikationsmerkmalen mit. Diese Anforderungen steuern die Interaktion zwischen einem Service-Client und einem Zielservice.

Qualifikationsmerkmale können für Servicekomponentenverweise, Schnittstellen und Implementierungen angegeben werden. Da die Deklaration von QoS-Werten außerhalb einer Implementierung (extern) erfolgt, können Sie diese Werte ändern, ohne die Implementierung zu ändern. Oder Sie können sie unterschiedlich einstellen, wenn verschiedene Instanzen der gleichen Implementierung in verschiedenen Kontexten verwendet werden.

Qualifikationsmerkmale besitzen die folgenden Kategorien:

- Transaktion: Regeln für den Typ von Transaktion
- Aktivitätssitzung: Regeln für das Verbinden der aktiven Sitzung
- Sicherheit: Regeln für Berechtigungen
- Asynchrone Zuverlässigkeit: Regeln für die asynchrone Nachrichtenübermittlung





---

## Kapitel 7. Module

Ein Modul ist eine Einsatzeinheit, die bestimmt, welche Artefakte in eine EAR-Datei (Enterprise Archive) gepackt werden. Die Komponenten in einem Modul werden aus Leistungsgründen zusammengefasst und können ihre Daten durch Verweise übergeben. Ein Modul kann als Scopingmechanismus betrachtet werden; das heißt, es setzt organisatorische Grenzen für Artefakte.

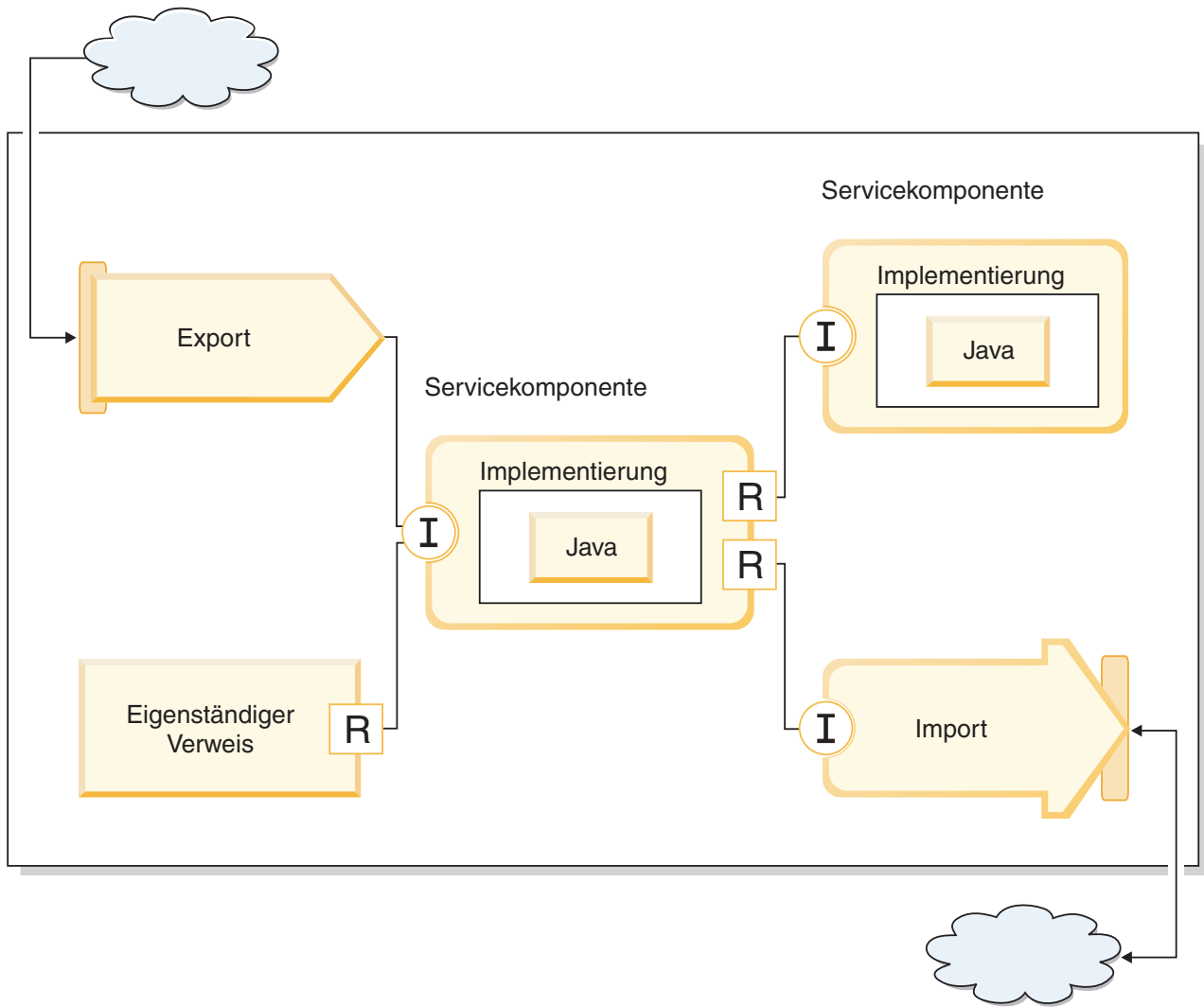
Ein Modul ist eine Kombination von Servicekomponenten, Importen und Exporten. Die Servicekomponenten, Importe und Exporte befinden sich in demselben Projekt- und Stammordner, in dem auch die Verkabelung enthalten ist, die die Komponenten und die Bindings verbindet, die für die Importe und Exporte erforderlich sind. Ein Modul kann auch Implementierungen und Schnittstellen enthalten, auf die von seinen Komponenten, Importen und Exporten verwiesen wird; diese können sich auch in anderen Projekten wie zum Beispiel einem Bibliotheksprojekt befinden.

Es gibt zwei Arten von Modulen. Einerseits gibt es Module mit der Bezeichnung Modul (wird manchmal auch als Geschäftsintegrationsmodul bezeichnet), die eine Auswahl vieler Komponententypen enthalten und oft zur Unterstützung eines Geschäftsprozesses verwendet werden. Andererseits gibt es Module mit der Bezeichnung Mediationsmodul, die maximal eine Komponente enthalten, eine Mediationsablaufkomponente, und zusätzlich keine oder mehrere Java-Komponenten, die die Mediationsablaufkomponente erweitern.

Warum gibt es zwei Modultypen? Der erste Modultyp ist vor allem für Geschäftsprozesse konzipiert. Ein Mediationsmodul ist wie ein Gateway zu vorhandenen externen Services, was in den Busarchitekturen eines Unternehmensservice häufig der Fall ist. Auf diese externen Services oder Exporte wird in einem Mediationsmodul durch Importe oder Serviceanbieter zugegriffen. Durch das Entkoppeln der Clientserviceanforderer von den Serviceanbietern mit Hilfe eines Mediationsablaufs steigt die Flexibilität und Ausfallsicherheit Ihrer Anwendungen, ein Ziel jeder serviceorientierten Architektur. Der Mediationsablauf kann zum Beispiel eingehende Nachrichten protokollieren, Nachrichten an einen bestimmten Service weiterleiten, der während der Ausführung ermittelt wird, oder Daten umwandeln, damit sie für die Übermittlung an einen anderen Service geeignet sind. Diese Funktionen können im Lauf der Zeit hinzugefügt oder verändert werden, ohne dass die Anfordererservices oder die Anbieterservices geändert werden müssen.

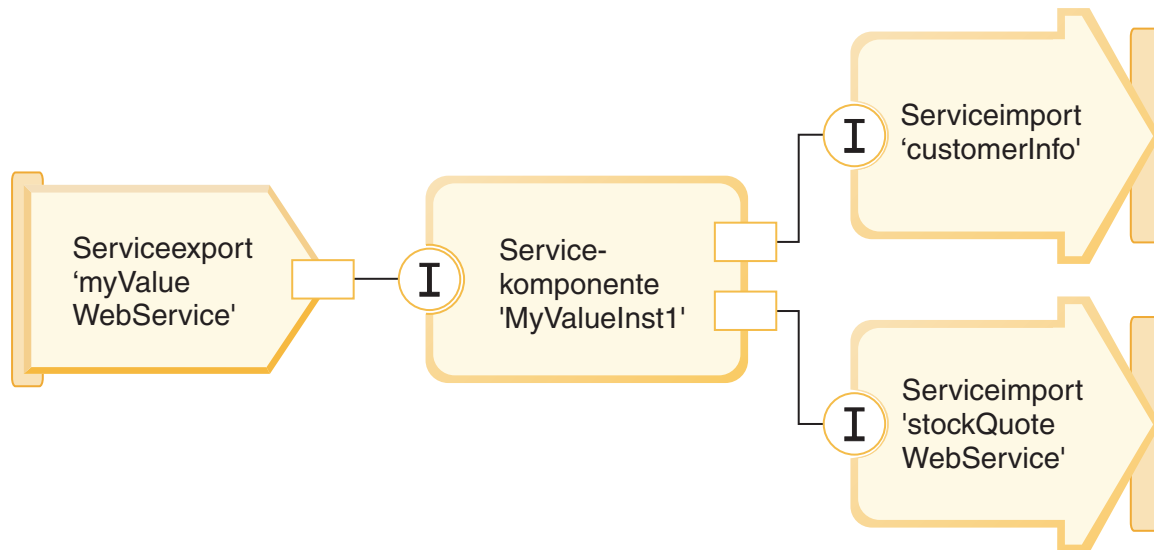
Ein Modul ist somit eine Serviceanwendung, die getestet und in WebSphere Process Server implementiert wurde. Ein Mediationsmodul ist eine Serviceanwendung, die getestet und entweder in WebSphere Process Server oder auf einem WebSphere Enterprise Service Bus-Server implementiert wurde. Beide Modultypen unterstützen Importe und Exporte.

Implementierungen, Schnittstellen, Geschäftsobjekte, Geschäftsobjektzuordnungen, Aufgabenbereiche, Beziehungen und andere Artefakte müssen häufig gemeinsam mit anderen Modulen verwendet werden. Eine Bibliothek ist ein Projekt, das zum Speichern dieser gemeinsamen genutzten Ressourcen verwendet wird.



Im folgenden Diagramm enthält das Modul einen Export, zwei Importe und eine Servicekomponente, die sie verwendet. Die gezeigte Verkabelung verbindet die Schnittstellen und Verweise.





Zu den Modul- und Mediationsmodulartefakten zählen:

- Moduldefinition: Definiert das Modul.
- Servicekomponenten: Definitionen der Services im Modul. Der Name einer Servicekomponente ist innerhalb eines Moduls eindeutig. Eine Servicekomponente kann jedoch auch einen beliebigen Anzeigenamen besitzen, der in der Regel eher für einen Benutzer nützlich ist.
- Importe: Definitionen von Importen, die Aufrufe von Services sind, die sich außerhalb des Moduls befinden. Importe besitzen Bindings, die im Abschnitt Importe und Exporte besprochen werden.
- Exporte: Definitionen von Exporten, die dazu verwendet werden, Komponenten anderen Aufrufenden verfügbar zu machen, die sich außerhalb dieses Moduls befinden. Exporte besitzen Bindings, die im Abschnitt Importe und Exporte besprochen werden.
- Verweise: Verweise von einer Komponente auf eine andere im Modul.
- Eigenständige Verweise: Referenzanwendungen, die nicht als SCA-Komponenten definiert sind (zum Beispiel JavaServer Pages), die es diesen Anwendungen ermöglichen, mit SCA-Komponenten zu interagieren. Pro Modul ist nur ein eigenständiges Verweisartefakt zulässig.
- Andere Artefakte: Zu diesen Artefakten gehören WSDL-Dateien, Java-Klassen, XSD-Dateien, BPEL-Prozesse usw.



---

## Kapitel 8. Importe und Exporte

Importe und Exporte definieren die externen Schnittstellen oder Zugriffspunkte eines Moduls. Importe geben Services außerhalb eines Moduls an, so dass sie aus dem Modul heraus aufgerufen werden können. Über Exporte können Komponenten ihre Services externen Clients zur Verfügung stellen. Für einen Import oder Export sind Binding-Informationen erforderlich. Es sind mehrere Bindings verfügbar, und es werden Informationen dazu angezeigt, welcher Typ für Ihre Anwendung geeignet ist.

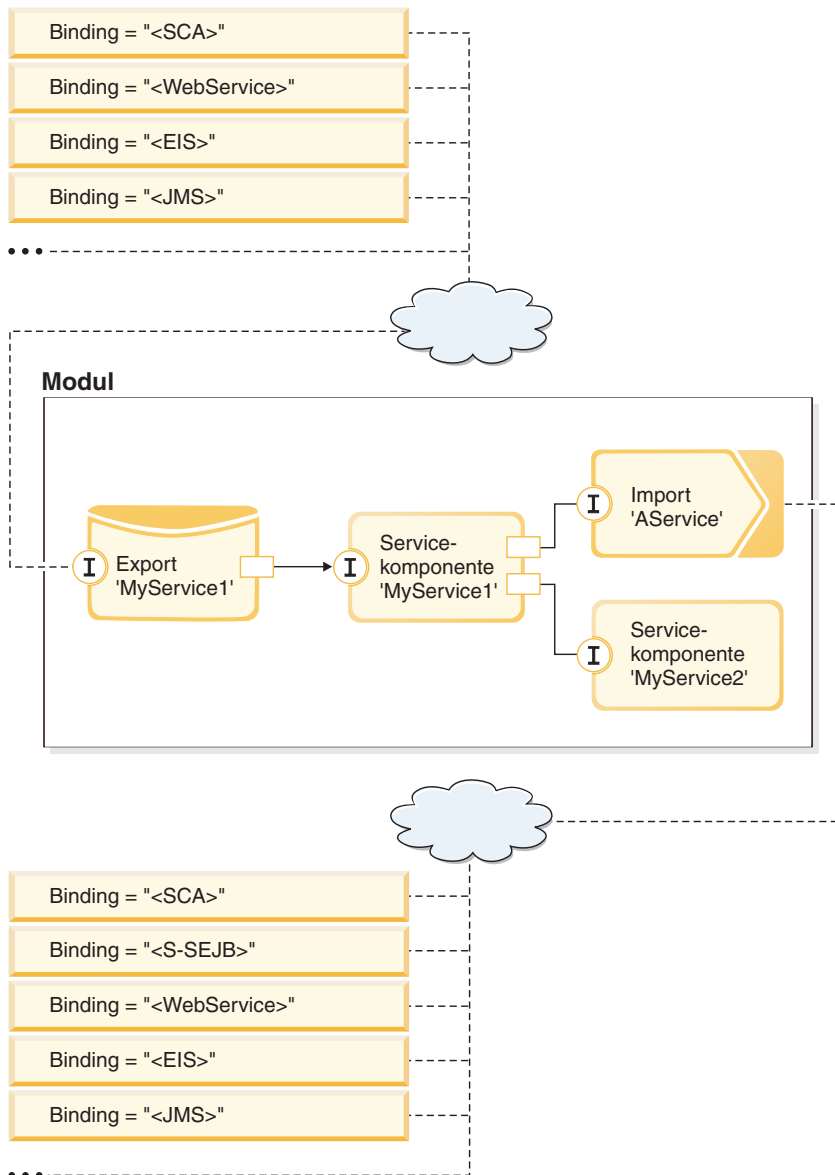
---

### Export- und Importbindingtypen für Services

Im- und Exporte erfordern Bindinginformationen, die das Transportmittel für die Daten aus den Modulen angeben. Ein Importbinding beschreibt die spezielle Methode, mit der ein externer Service an eine Importkomponente gebunden wird. Ein Exportbinding beschreibt die spezielle Methode, mit der die Services eines Moduls für Clients verfügbar gemacht werden.

Das SCA- oder Standardbinding ermöglicht die Kommunikation Ihres Service mit anderen Services in anderen Modulen. Ein Import mit einem SCA-Binding ermöglicht Ihnen den Zugriff auf einen Service in einem anderen Modul. Ein Export mit einem SCA-Binding ermöglicht Ihnen, einen Service anderen Modulen anzubieten. Ein Web-Service-Importbinding ermöglicht Ihnen das Binden eines externen Web-Service an einen Import. Ein Web-Service-Exportbinding ermöglicht Ihnen die Bereitstellung eines Service für externe Clients in Form von Web-Services. Der Erkennungsassistent für Unternehmensservices erstellt Importe und Exporte, die einen Service auf einem EIS-System darstellen. Die erstellten Bindings sind vom Typ EIS oder JMS (Java Message Service). Ein EIS-Binding bietet synchrone Kommunikation mit dem Service auf dem EIS-System. Ein JMS-Binding wird in der Regel in Interaktionen mit großen EIS-Systemen genutzt, bei denen eine asynchrone Kommunikation über eine Nachrichtenwarteschlange von kritischer Bedeutung für die Zuverlässigkeit ist. Ein Import (jedoch kein Export) kann außerdem ein Session-EJB-Binding ohne Status (stateless) haben.

Der Assembly-Editor listet die unterstützten Bindings auf und vereinfacht ihre Erstellung, wenn Sie einen Import oder einen Export erstellen möchten. Eine Eigenschaftensicht im Assembly-Editor zeigt die Bindinginformationen jedes Imports bzw. Exports.



## Geeignete Bindings auswählen

In diesem Abschnitt wird besprochen, ob ein bestimmtes Binding für die Anforderungen Ihrer Anwendung besser geeignet sein könnte.

Die in WebSphere Integration Developer verfügbaren Bindings bieten eine Reihe an Möglichkeiten. Diese Liste hilft Ihnen dabei, zu erkennen, wann ein Bindingtyp für die Anforderungen einer Anwendung geeigneter ist als ein anderer Bindingtyp.

Nehmen wir ein Binding des Typs SCA, wenn die folgenden Faktoren gegeben sind:

- Alle Services sind in den Modulen von WebSphere Integration Developer enthalten; das bedeutet, dass es keine externen Services gibt
- Die Leistung ist wichtig
- Die Module sind eng verbunden

Nehmen wir ein Binding des Typs Web-Services, wenn die folgenden Faktoren gegeben sind:

- Sie müssen auf einen externen Service über das Internet zugreifen oder einen Service über das Internet bereitstellen
- Die Services sind lose verbunden
- Das Protokoll eines externen Service, auf den Sie zugreifen bzw. eines Service, den Sie bereitstellen möchten, ist SOAP/HTTP oder JMS/HTTP

Nehmen wir ein Binding des Typs EIS, wenn die folgenden Faktoren gegeben sind:

- Sie müssen mit einem Ressourcenadapter auf einen Service auf einem EIS-System zugreifen
- Die Leistung ist wichtiger als die Zuverlässigkeit; das heißt, die synchrone Datenübertragung ist der asynchronen vorzuziehen

Nehmen wir ein Binding des Typs JMS, wenn die folgenden Faktoren gegeben sind:

- Sie müssen auf ein Nachrichtenübermittlungssystem zugreifen
- Die Services sind lose verbunden
- Die Zuverlässigkeit ist wichtiger als die Leistung; das heißt, die asynchrone Datenübertragung ist der synchronen vorzuziehen

Nehmen wir ein Binding des Typs Session-EJB ohne Status, wenn die folgenden Faktoren gegeben sind:

- Das Binding ist für einen importierten Service, der selbst eine EJB ist
- Der importierte Service ist lose verbunden
- Der Status der EJB ist nicht wichtig



---

## Kapitel 9. Serviceimplementierungstypen

Serviceimplementierungstypen sind die Implementierungen der Servicekomponenten.

Die Standardimplementierungen der Services werden in diesem Abschnitt beschrieben. Diese Implementierungen werden im Assembly-Editor bzw. innerhalb von BPEL-Prozessen als Services angezeigt.

---

### Java-Objekte

Eine Implementierung einer Komponente in Java wird als Java-Objekt bezeichnet.

Eine in Java geschriebene Komponente ist eine gängige Implementierung. Diese Implementierung wird manchmal als "einfaches altes Java-Objekt" (engl.: POJO - plain old Java object) bezeichnet. Im Allgemeinen besitzt eine solche Implementierung einen WSDL-Schnittstellentyp, obwohl sie auch eine Java-Schnittstelle haben könnte. Wenn mehrere Schnittstellen angegeben sind, können Sie WSDL-Schnittstellen nicht mit Java-Schnittstellen vermischen. Sie können jedoch eine mit einer Gruppe von WSDL-Schnittstellen erstellte Anwendung mit einer Anwendung mit einer Gruppe von Java-Schnittstellen "verknüpfen". Wie dies geschieht, zeigt ein Beispiel, das in der Beispielgalerie der Sicht 'Willkommen' zu finden ist.

Bei der Arbeit mit einem Java-Objekt bleibt Ihnen der Code im Kontext der Editoren verborgen.

Ein Java-Objekt kann nicht in einem Mediationsmodul verwendet werden. Es kann entweder in WebSphere Process Server oder in einen WebSphere Enterprise Service Bus-Server integriert werden.

---

### BPEL-Prozess

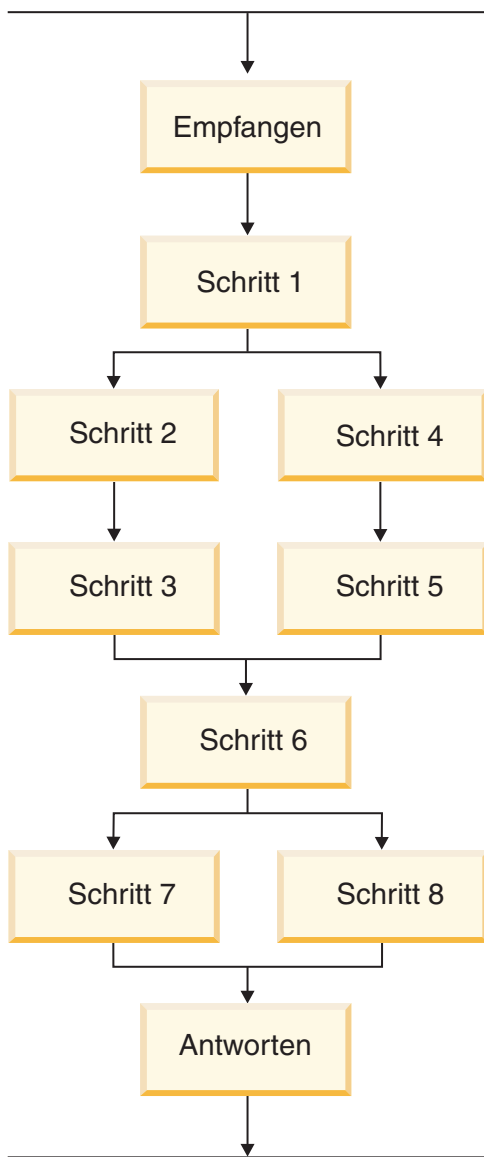
Eine BPEL-Prozesskomponente implementiert einen Geschäftsprozess.

Implementierungssprache sind der Industriestandard Business Process Execution Language for Web Services (BPEL4WS) sowie die zugehörigen IBM Erweiterungen. Ein BPEL-Prozess implementiert einen Service mit potenziell langer Laufzeit durch Verwendung mehrerer elementarer Services. Ein im Prozesseditor erstellter BPEL-Prozess verfügt über folgende Funktionalität:

- Beschreiben der Koordination anderer Services mit Hilfe von Steuerungsflussdiagrammen
- Verwenden von Variablen zur Verwaltung des Prozess-Status
- Verwenden hoch entwickelter Fehlerbehandlungsfunktionen
- Unterstützung für asynchrone Ereignisse
- Korrelieren eingehender Anforderungen mit dem richtigen Exemplar eines bestimmten Prozesses durch Verwendung von Korrelationsgruppen zur Markierung derjenigen Geschäftsdaten innerhalb einer Anforderung, welche das Exemplar identifizieren (z. B. eine Kunden-ID)
- Bereitstellen erweiterter Transaktionen durch hoch entwickelte Kompensationsunterstützung

Neben diesen BPEL-Standardelementen wird BPEL von WebSphere Integration Developer zudem um eine Human Task-Unterstützung zur Einbindung von Personen in einen Prozess erweitert. Diese Erweiterung könnte zum Beispiel einem Prozess die Bestimmung hinzufügen, dass eine Person ein Darlehen genehmigen muss.

Der Prozesseditor arbeitet mit graphischen Darstellungen von BPEL-Konstrukten zur raschen und einfachen Erstellung Ihres Geschäftsprozesses.



Ein BPEL-Prozess kann nicht in einem Mediationsmodul verwendet werden. Er kann nur in WebSphere Process Server implementiert werden.

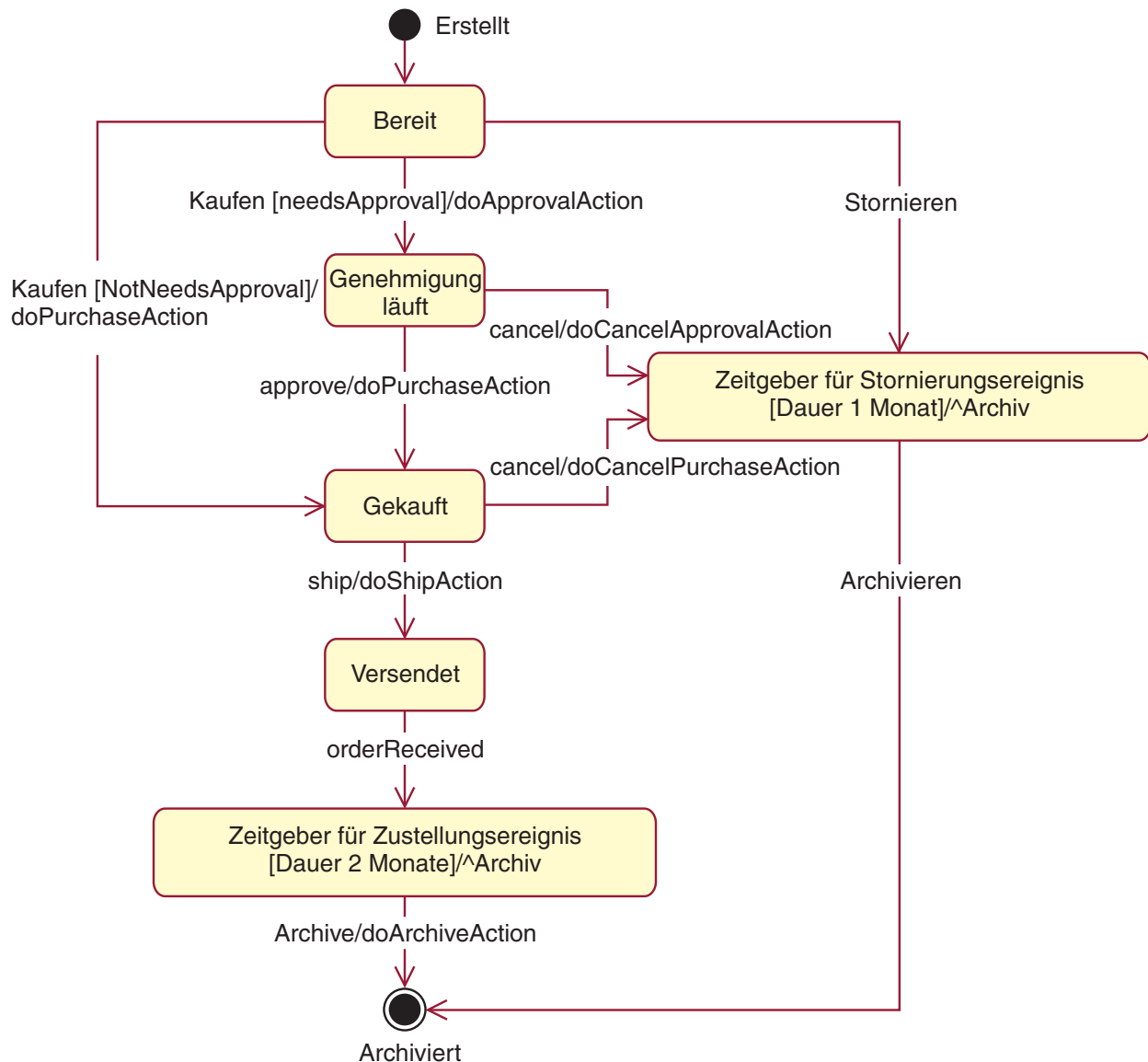
---

## Statusmaschinen

Eine Statusmaschine ist eine alternative Möglichkeit zum Erstellen eines Geschäftsprozesses. Eine Statusmaschine ist für Prozesse geeignet, die eher zum Ändern eines Status als zum Ändern eines Steuerungsflusses vorgesehen sind. Ein Status definiert, welche Funktionen ein Artefakt zu einem bestimmten Zeitpunkt ausführen kann. Eine Statusmaschine ist eine Implementierung eines solchen Statussatzes.

Statusmaschinen sind eine allgemeine Methode zur Anzeige eines Satzes gegenseitig abhängiger Status in einem Prozess. Eine vertraute Art von Statusmaschine ist zum Beispiel ein Getränkeautomat. Wenn Sie einige Münzen in den Apparat einwerfen, erhalten Sie zusammen mit dem Getränk (bei ordnungsgemäßen Betrieb) das exakte Wechselgeld zurück. Dazu ermittelt die Statusmaschine mechanisch die Münzen, die Ihnen abhängig von den eingeworfenen Münzen wieder herauszugeben sind. Im folgenden Diagramm wird eine typische Statusmaschine als vom Statusmaschineneditor erstellte Statusmaschine dargestellt. In der Statusmaschine wird ein Artikel gekauft und an den Kunden geliefert.





Eine Statusmaschine kann nicht in einem Mediationsmodul verwendet werden. Sie kann nur in WebSphere Process Server implementiert werden.

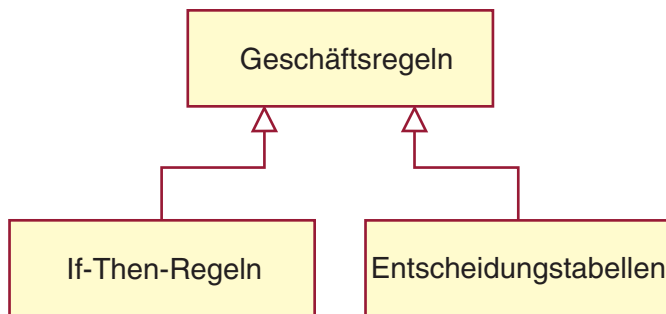
## Geschäftsregeln

Geschäftsregeln ergänzen Geschäftsprozesse und Statusmaschinen. Im Fall einer Bedingung mit einer Variablen kann zum Beispiel eine Geschäftsregel den Wert der Variablen während der Ausführung verändern. Da eine Geschäftsregel unter Verwendung einer graphisch orientierten Programmiersprache erstellt wurde, kann sie Entscheidungen auf der Basis eines Kontexts treffen. Die Entscheidung kann einfach oder komplex sein. Geschäftsregeln sind nicht prozedural, und die Regeln können unabhängig von einer Anwendung geändert werden.

Geschäftsregeln bestimmen das Ergebnis eines Prozesses auf der Basis eines Kontexts. Geschäftsregeln kommen in täglichen Geschäftssituationen zum Einsatz, um nach Maßgabe einer definierten Gruppe von Umständen eine Entscheidung zu treffen. Diese Entscheidung kann zur Abdeckung aller möglichen Umstände zahlreiche Regeln erforderlich machen. Geschäftsregeln innerhalb eines Geschäftsprozesses geben Anwendungen die Möglichkeit, schnell auf Änderungen von Geschäftsbedingungen zu reagieren. In einem Versicherungsunternehmen könnte eine Geschäftsregel zur Genehmigung einer KFZ-Versiche-

rung für einen Antragsteller wie folgt lauten: Wenn der Antragsteller männlich und über 25 Jahre alt ist und die KFZ-Kategorie Sportwagen ist und die Person über die letzten fünf Jahre bei uns versichert war, dann ist der Antrag auf Versicherung zu einer Monatsprämie von 100 EUR zu genehmigen.

WebSphere Integration Developer bietet eine Reihe von Methoden zur Erstellung von Geschäftsregeln an. Sie können If-Then-Regeln oder Entscheidungstabellen erstellen, die sämtlich das Ergebnis Ihres Prozesses formen. Beachten Sie, dass diese Regeln unabhängig von dem eigentlichen Prozess sind. Das heißt, Sie können die Regeln jederzeit ändern, ohne den Prozess erneut erstellen zu müssen. Abhängig vom Standort Ihres Unternehmens könnten Sie zum Beispiel eine Regel wie die folgende haben: Wenn das Datum zwischen dem 26. Dezember und dem 1. Januar liegt, ist ein Urlaubsnachsaisonrabatt von 20% anzubieten. Sollten sich die Verkaufszahlen weiterhin nur zäh entwickeln, könnten Sie jederzeit den Rabatt auf 40% ändern.



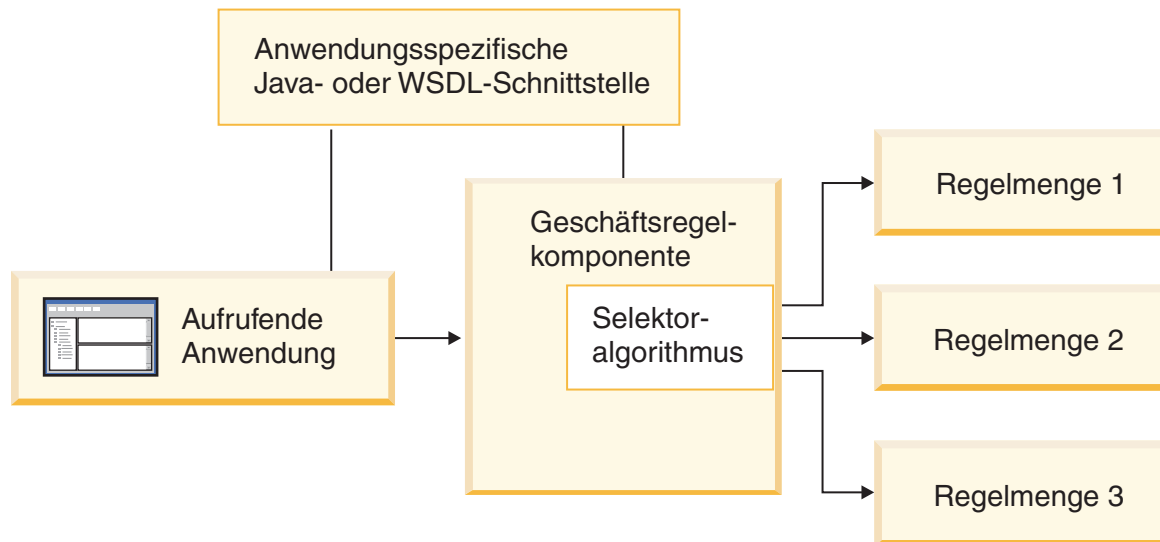
Geschäftsregeln können nicht in einem Mediationsmodul verwendet werden. Sie können nur in WebSphere Process Server implementiert werden.

---

## Selektoren

Integrierte Anwendungen bieten viele Möglichkeiten zur Interaktion. Ein Selektor leitet eine Operation von einer Clientanwendung an eine von mehreren möglichen Komponenten zur Implementierung weiter.

Die Basis für die Weiterleitung an eine Komponente ist das Datum. Ein Beispiel für eine Weiterleitung auf der Basis des Datums könnte wie folgt lauten: Zwei Wochen vor Schulbeginn ist ein Sonderpreis zum Ferienende für schulbezogene Verkaufsartikel anzubieten. Unternehmen können viele solche Weiterleitungen auf der Basis des Datums verwenden. Ein Selektor entscheidet, welche Route einer anderen auf der Basis des Datums während der Ausführung vorzuziehen ist. Wenn der Zeitpunkt zum Beispiel kurz vor Schulbeginn liegt, sollte das oben erwähnte Ferienendeangebot aufgerufen werden. Liegt der Zeitpunkt jedoch in der letzten Phase vor den Ferien, könnte ein Angebot erfolgen, das Kinder auf den Sommer einstimmt.

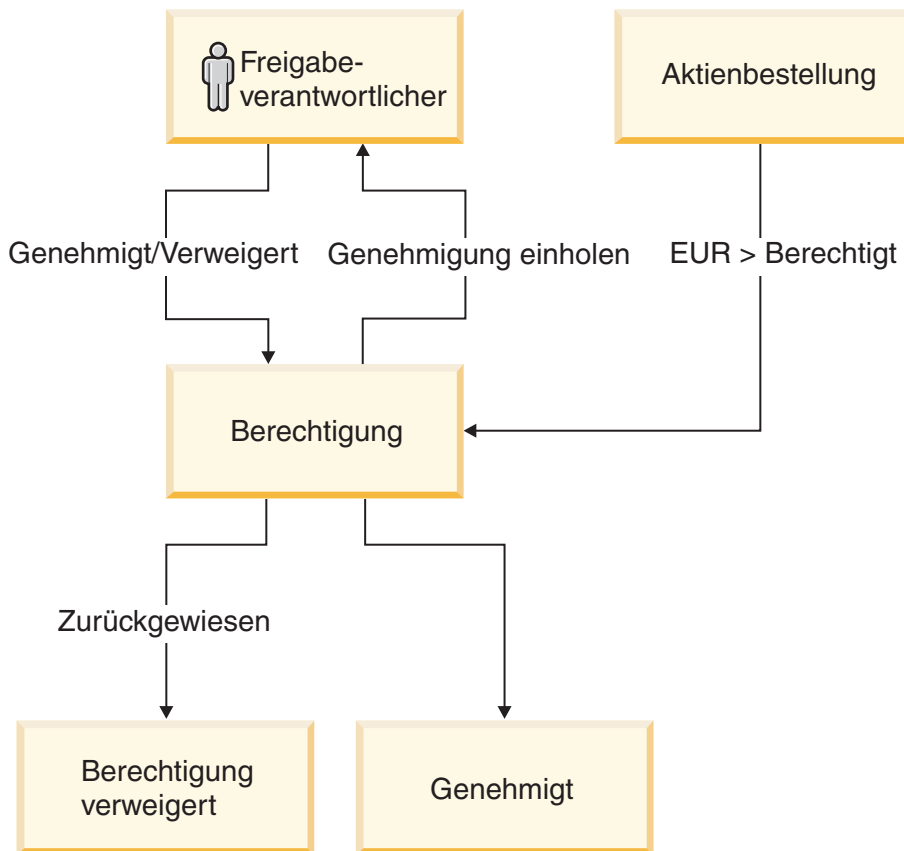


Ein Selektor kann nicht in einem Mediationsmodul verwendet werden. Er kann nur in WebSphere Process Server implementiert werden.

## Human Task

Eine Human Task-Komponente implementiert eine Task, die von einer Person ausgeführt wird. Sie stellt die Beteiligung einer Person an einem Geschäftsprozess dar.

Es gibt Fälle, in denen Personen in einen Geschäftsprozess eingreifen müssen. Ein Kunde möchte zum Beispiel einen Artikel kaufen, der über seinem Kreditrahmen liegt. Eine Human Task bietet Ihnen die Möglichkeit, eine Geschäftsregel, die den Kunden an der Realisierung seines Kaufs hindert, durch einen persönlichen Eingriff außer Kraft zu setzen. Eine Human Task kann Attribute aufweisen. Dazu gehört zum Beispiel die Einstellung des Eigners der Task sowie die Angabe eines Eskalationsprozesses für den Fall, dass die angegebene Person nicht verfügbar ist. Die Human Task-Komponente erkennt, dass für viele Prozesse ein manueller Eingriff für Tasks wie Überprüfen, Suchen und Genehmigen erforderlich ist.



Eine Human Task kann nicht in einem Mediationsmodul verwendet werden. Sie kann nur in WebSphere Process Server implementiert werden.

## Schnittstellenzuordnung

Eine Schnittstellenzuordnung löst Unterschiede zwischen den Schnittstellen der interagierenden Komponenten auf.

Unterschiede zwischen Schnittstellen in den Komponenten, die miteinander interagieren müssen, kommen häufig vor. Diese Unterschiede entstehen, weil in WebSphere Integration Developer oft Komponenten assembliert werden, die für unterschiedliche Anwendungen erstellt wurden. Ihre Wiederverwendung zum Erstellen einer neuen Anwendung ist eine der Stärken von WebSphere Integration Developer, da andernfalls an ähnlichen Komponenten Änderungen am Programmcode vorgenommen werden müssten. In der Regel müssen Sie aber noch einige Anpassungen vornehmen.

Zwei Komponenten können beispielsweise über Methoden verfügen, die im Wesentlichen dieselbe Aktion ausführen, aber verschiedene Namen besitzen, wie zum Beispiel 'getCredit' und 'getCreditRating'. Sie können auch unterschiedliche Operationsnamen besitzen und die Parameterarten der Operationen können verschieden sein. Eine Schnittstellenzuordnung ordnet die Operationen und Parameter dieser Methoden zu, so dass die Unterschiede aufgelöst werden und die zwei Komponenten interagieren können. Eine Schnittstellenzuordnung ist wie eine Brücke zwischen den Schnittstellen von zwei Komponenten, die es ihnen ermöglicht, trotz der Unterschiede eine Verbindung zueinander herzustellen.

Eine Schnittstellenzuordnung existiert unabhängig von den Komponenten, die sie verwenden; das bedeutet, dass die Komponenten selbst nicht verändert werden müssen.

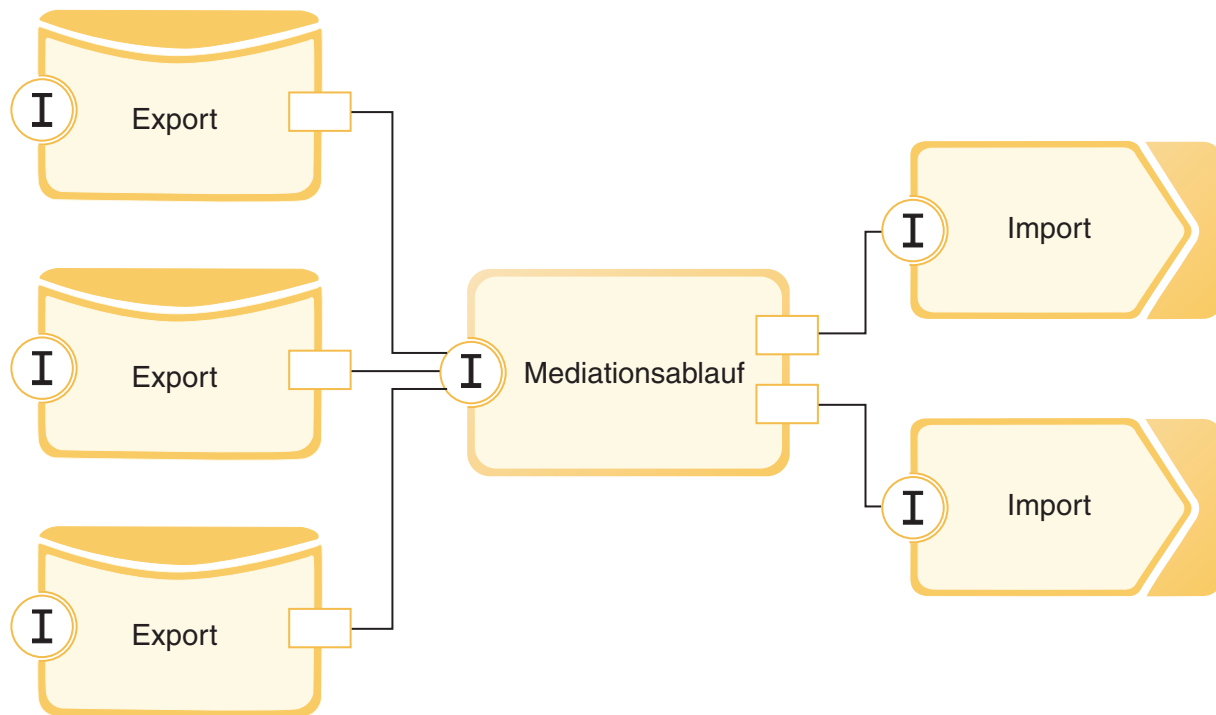
Eine Schnittstellenzuordnung kann nicht in einem Mediationsmodul verwendet werden. Sie kann nur in WebSphere Process Server implementiert werden.

## Mediationsablauf

Mediation ist eine Art der Vermittlung bzw. des dynamischen Eingreifens zwischen Services. Ein Mediationsablauf implementiert eine Mediation.

Die Mediation verfügt über eine Reihe nützlicher Funktionen. Sie können die Mediation zum Beispiel nutzen, wenn Sie Daten von einem Service in ein zulässiges Format für einen anderen Service bringen möchten. Mit der Protokollierung können Sie Nachrichten von einem Service aufzeichnen, bevor sie an den nächsten Service gesendet werden. Die Weiterleitung ermöglicht es Ihnen, Daten von einem Service an einen geeigneten Service weiterzuleiten, der mit Hilfe des Mediationsablaufs ermittelt wurde. Eine Mediation arbeitet unabhängig von den Services, die sie verbindet. Eine Mediation wird im Assembly-Editor als eine Mediationsablaufkomponente zwischen Exporten und Importen angezeigt.

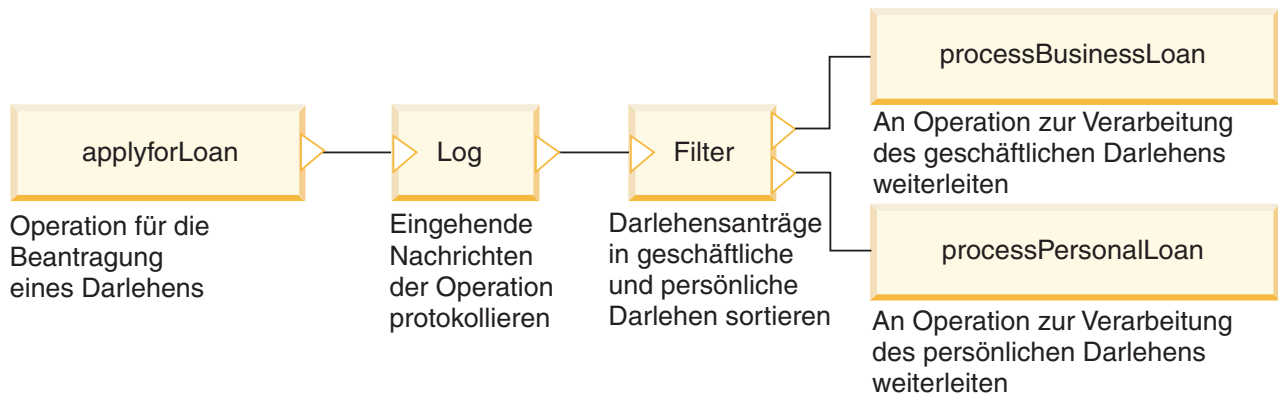
Im folgenden Diagramm senden drei Serviceanforderer bzw. Exporte ihre Ausgabedaten an die Schnittstelle der Mediationsablaufkomponente. Die Mediationsablaufkomponente leitet die entsprechenden Daten anschließend an die zwei Serviceanbieter bzw. Importe weiter.



Ein Mediationsablauf ist ein flussähnliches Konstrukt, das mit Hilfe des Mediationsablauf-Editors erstellt wird. Wenn Sie eine Mediationsablaufkomponente im Assembly-Editor auswählen, wird der Mediationsablauf-Editor gestartet. Im Mediationsablauf-Editor wird eine Operation von einem Service, dem Serviceanforderer oder Export, der Operation eines anderen Services, des Serviceanbieters oder Imports, mit den vom Mediationsablauf-Editor zur Verfügung gestellten Funktionen zugeordnet. Diese Funktionen werden Mediationsbasiselemente genannt und sind in einem Mediationsablauf wie im folgenden Diagramm dargestellt verknüpft. Mediationsbasiselemente werden von IBM zur Verfügung gestellt oder können von Ihnen als benutzerdefinierte Basiselement erstellt werden. Mediationsbasiselemente können sowohl als Nachrichteninhalt und Nachrichtenkontext (der Kontext ist dann eine bindungsspezifische Information wie SOAP oder JMS-Header) oder als benutzerdefinierte Eigenschaften verwendet werden.

In dem Diagramm, das auf eine Operation folgt und 'applyforLoan' genannt wird, wird eine Nachricht zuerst an das Protokollierungsbasiselement 'Log' gesendet, das die Nachricht aufzeichnet. 'Log' sendet

die Nachricht an das Basiselement 'Filter', das die Nachricht je nach Nachricht entweder an die Operation 'processBusinessLoan' oder an die Operation 'processPersonalLoan' weiterleitet.



Wie bereits im Abschnitt Module erläutert, gibt es für eine Mediationsablaufkomponente ein Mediationsmodul. Es kann maximal eine Mediationsablaufkomponente oder eine Komponente und mehrere Java-Komponenten enthalten, die die Mediationsablaufkomponente erweitern. Ein Mediationsmodul kann entweder in WebSphere Process Server oder in einen WebSphere Enterprise Service Bus-Server integriert werden.

---

## Eigenständige Verweise

Eigenständige Verweise sind Verweise auf Anwendungen, die nicht als SCA-Komponenten definiert sind (zum Beispiel JavaServer Pages oder Servlets). Eigenständige Verweise ermöglichen diesen Anwendungen das Interagieren mit SCA-Komponenten (SCA, Service Component Architecture).

Eigenständige Verweise verfügen weder über eine Schnittstelle noch über eine Implementierung (da die Implementierung außerhalb des Bereichs des Moduls ist). Ein Modul kann keine eigenständigen Verweise oder ein eigenständiges Verweisartefakt besitzen. Eigenständige Verweise bieten den praktischen Vorteil, dass sie das Verwenden bereits vorhandener Anwendungen zusammen mit den SCA-Komponenten ermöglichen, die in WebSphere Integration Developer erstellt wurden.

Eigenständige Verweise können in einem Mediationsmodul verwendet werden. Sie können entweder in WebSphere Process Server oder in einen WebSphere Enterprise Service Bus-Server integriert werden.

---

## Kapitel 10. Zugehörige Informationen

In diesem Abschnitt werden mehrere Themen besprochen, die sich auf die Architektur beziehen.

Diese Themen stellen weitere Informationen zur Verfügung, die sich auf die Architektur des Produkts beziehen.

- „Services für die Verwendung mit .NET-Services entwickeln“
- „Bidirektionale Unterstützung“

### Services für die Verwendung mit .NET-Services entwickeln

Falls Sie beabsichtigen, Services zu entwickeln, die mit .NET-Services verwendet werden sollen, müssen Sie einige besondere Überlegungen berücksichtigen. Dabei stehen zwei Aspekte im Vordergrund: Das Importieren von WSDL-Dateien, die in einer .NET-Entwicklungsumgebung entwickelt wurden, in die Entwicklungsumgebung von WebSphere Integration Developer, und das Exportieren von WSDL-Dateien aus WebSphere Integration Developer, so dass Sie diese mit .NET-Services verwenden können. Der Hauptunterschied zwischen den beiden Entwicklungsumgebungen ist die Tatsache, dass die .NET-Umgebung Inline-Schemata verwendet und WebSphere Integration Developer nicht. Inline-Schemata sind eine Möglichkeit zum Einschließen des Schemas in eine WSDL-Datei; somit ist die Angabe überflüssig, dass das Schema als separate Datei importiert werden muss. Hilfeinformationen für beide Fälle finden Sie in WebSphere Integration Developer.

Inline-Schemata im Abschnitt 'Schnittstellen erstellen' zeigt auf, wie WSDL-Dateien importiert werden müssen, die Inline-Schemata enthalten, wie dies bei .NET-WSDL-Dateien der Fall ist.

Creating proxies to work with .NET services im Abschnitt 'Geschäftsservice assemblieren' zeigt auf, wie Services so extern veröffentlicht werden, dass sie mit .NET-Services funktionieren. Dieses Thema behandelt vor allem die Erstellung von Proxys, die mit .NET-Services arbeiten.

### Bidirektionale Unterstützung

WebSphere Integration Developer arbeitet in einer Mehrsprachenumgebung. Das bedeutet, Daten in unterschiedlichen Sprachen können bearbeitet und angezeigt werden. Diese Unterstützung umfasst auch einige Sprachen, die bidirektionale Scripts besitzen (zum Beispiel Arabisch oder Hebräisch). Diese Sprachen werden von rechts nach links geschrieben, während die Ziffern und eingebettete Textsegmente aus dem lateinischen Zeichensatz (zum Beispiel Kyrillisch, Griechisch, etc) in diesem Text von links nach rechts geschrieben werden.

In Overview of bidirectional script support in IBM WebSphere Integration Developer wird die Unterstützung für bidirektionale Sprachen besprochen; unter anderem werden die erforderliche Konfiguration, eine Reihe technischer Aspekte bei der Verwendung der Unterstützung sowie die Einschränkungen der Unterstützung erläutert.

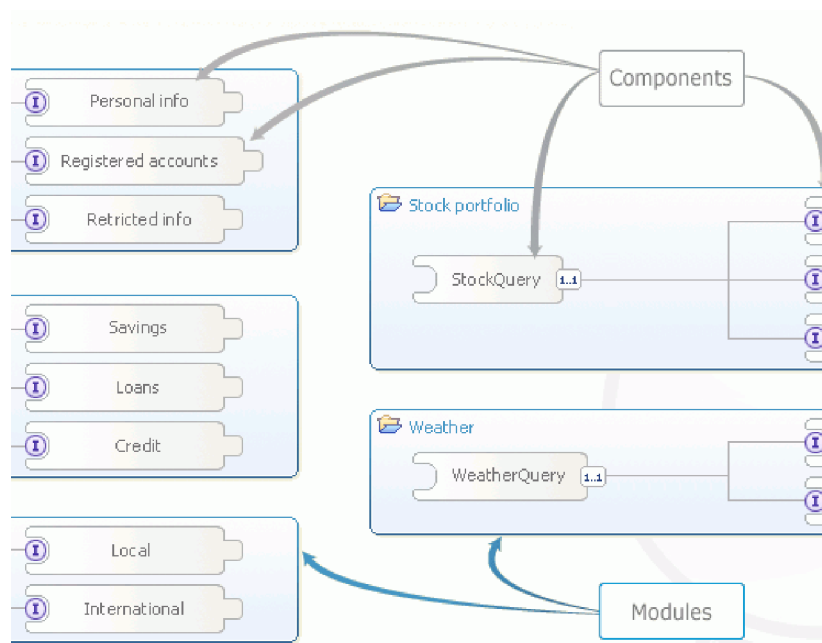




## Kapitel 11. Sicht 'Willkommen' - Übersicht

Der Übersichtsabschnitt der Sicht 'Willkommen' bietet den allgemeinsten Blick auf die Tools und Funktionen im Produkt. In der Übersicht dienen Grafiken zur Darstellung der Funktionen der wichtigsten Tools im Produkt.

Der Übersichtsabschnitt der Sicht 'Willkommen' bietet eine Möglichkeit, sich vor dem Einstieg in die eigentliche Arbeit bequem und schnell mit dem Produkt vertraut zu machen. Die meisten Benutzer, die das Produkt noch nicht kennen, sollten an diesem Punkt starten. Durch Auswahl eines Symbols in der Übersicht wird eine graphische Darstellung zu einem Tool oder einem Thema angezeigt. Im folgenden Diagramm werden zum Beispiel die Beziehungen von Komponenten und Modulen gezeigt. Sie können außerdem eine audiovisuelle Tour durch das Produkt starten.



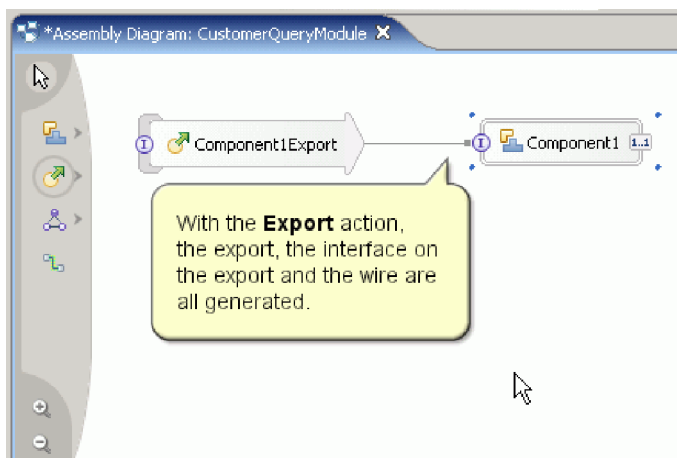


---

## Kapitel 12. Lernprogramme in der Sicht 'Willkommen'

Die Lernprogramme geben Ihnen die Möglichkeit, die Tools auf fehlersichere Weise in Aktion zu sehen. Sie wählen ein Lernprogramm aus und beobachten, wie es einige Tasks ausführt, wie es ein Benutzer tun würde.

Wenn Sie sich mit dem Produkt auf der konzeptionellen Ebene vertraut gemacht haben, können Sie die Tools ausprobieren, indem Sie ein Lernprogramm starten. Jedes Lernprogramm ist eine Art Filmvorführung zu einem bestimmten Tools. Während Sie sich das Lernprogramm ansehen, beschreiben Textblasen, was Sie gerade sehen. Die Lernprogramme von WebSphere Integration Developer befinden Sie im Abschnitt 'Watch and Learn' der Lernprogrammsammlung in der Sicht 'Willkommen'.





---

## Kapitel 13. Beispiele in der Sicht 'Willkommen'

Die mitgelieferten Beispiele geben Ihnen eine praktische Anleitung in einer fehlersicheren Umgebung zur Verwendung von WebSphere Integration Developer zur Entwicklung von Servicelösungen.

Die Beispiele befinden sich in der Beispielgalerie der Sicht 'Willkommen'. Die Beispiele stellen verschiedene Optionen zur Auswahl. Sie können sie selbst anhand von schrittweisen Anleitungen erstellen, oder Sie können sie erstellen lassen.

Die Beispiele haben unterschiedlichen Inhalt und Umfang. Ein Technologiebeispiel demonstriert die Funktionsweise eines bestimmten Tools zur Ausführung einer bestimmten Task. Ein Anwendungsbeispiel zeigt die Verwendung mehrerer Tools zur Erzielung komplexerer Ergebnisse. Ein Szenario ist ein längeres Beispiel, das die Erstellung einer großen Anwendung mit zahlreichen Tools demonstriert. Für jedes Beispiel ist der zur Erstellung erforderliche Zeitaufwand angegeben.

### 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](#)

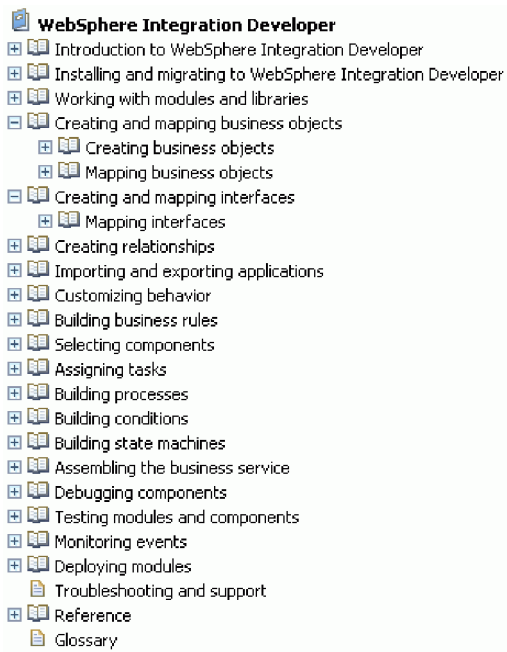


---

## Kapitel 14. Informationszentrum

Das Informationszentrum bietet vollständige Informationen zum Produkt. Neben Lerntexten (Tutorials) und Beispielen finden Sie für jedes einzelne Tool für WebSphere Integration Developer auch zusätzliche Informationen zu Konzepten und Tasks sowie Referenzinformationen.

Die Navigation des Informationszentrums bietet Ihnen einen raschen Zugriff auf die gewünschten Informationen. Sie können darüber hinaus das gesamte Informationszentrum durchsuchen. Wählen Sie unter den Suchergebnissen aus, um verschiedene Abschnitte mit detaillierten Informationen zum Thema sowie Links zu zugehörigen Informationen anzuzeigen.







---

## Bemerkungen

Die Verwendung der im vorliegenden IBM Produkt enthaltenen XDoclet-Dokumentation wurde genehmigt und wird durch den folgenden Copyrightvermerk geschützt: Copyright (c) 2000-2004, XDoclet Team. Alle Rechte vorbehalten.

Bestimmte Teile basieren auf der Veröffentlichung *Design Patterns: Elements of Reusable Object-Oriented Software*, von Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides, Copyright (c) 1995 durch Addison-Wesley Publishing Company, Inc. Alle Rechte vorbehalten.

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern oder Regionen nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in dieser Dokumentation beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Dokumentation ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing  
IBM Europe, Middle East & Africa  
Tour Descartes  
2, avenue Gambetta  
92066 Paris La Defense  
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hierin enthaltenen Angaben werden in regelmäßigen Zeitabständen aktualisiert. Diese Änderungen werden in Überarbeitungen bzw. neuen Editionen der Veröffentlichung bekannt gegeben. IBM kann jederzeit ohne vorherige Ankündigung Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in dieser Veröffentlichung auf Websites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

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

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in der Dokumentation aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung sowie der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Die oben genannten Erklärungen bezüglich der Produktstrategien und Absichtserklärungen von IBM stellen die gegenwärtige Absicht der IBM dar, unterliegen Änderungen oder können zurückgenommen werden, und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

(C) (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus IBM Corp. Musterprogrammen der IBM Corp. abgeleitet. (C) Copyright IBM Corp. 2000, 2005. Alle Rechte vorbehalten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farabbildungen.

## **Informationen zur Programmierschnittstelle**

Werden Informationen zur Programmierschnittstelle bereitgestellt, ermöglichen Ihnen diese das Erstellen von Anwendungssoftwareprogrammen mit Hilfe dieses Programms.

Allgemeine Programmierschnittstellen ermöglichen Ihnen das Schreiben von Anwendungssoftwareprogrammen, die die Services der Tools des vorliegenden Programms nutzen.

Diese Informationen enthalten möglicherweise auch Diagnose-, Änderungs- und Optimierungsinformationen. Diese Informationen werden bereitgestellt, um Ihnen die Behebung von Fehlern in Ihren Anwendungssoftwareprogrammen zu erleichtern.

Achtung: Diese Diagnose-, Änderungs- und Optimierungsinformationen dürfen nicht als Programmierschnittstelle verwendet werden, da sie jederzeit geändert werden können.

## **Marken und Servicemarken**

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







Gedruckt in Deutschland

SC12-3568-01

