

IBM DB2 Universal Database



IBM Video Central für e-business Lernprogramm

Version 8.1

IBM DB2 Universal Database



IBM Video Central für e-business Lernprogramm

Version 8.1

Anmerkung:

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter Anhang C, „Bemerkungen“ auf Seite 91 gelesen werden.

- Die IBM Homepage finden Sie im Internet unter: **ibm.com**
- IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation.
- Das e-business Symbol ist eine Marke der International Business Machines Corporation
- Infoprint ist eine eingetragene Marke der IBM.
- ActionMedia, LANDesk, MMX, Pentium und ProShare sind Marken der Intel Corporation in den USA und/oder anderen Ländern.
- C-bus ist eine Marke der Corollary, Inc. in den USA und/oder anderen Ländern.
- Java und alle Java-basierenden Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- Microsoft Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.
- PC Direct ist eine Marke der Ziff Communications Company in den USA und/oder anderen Ländern.
- SET und das SET-Logo sind Marken der SET Secure Electronic Transaction LLC.
- UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.
- Marken anderer Unternehmen/Hersteller werden anerkannt.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Universal Database IBM Video Central for e-business tutorial, Version 8.1,

herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2001, 2002.

© Copyright IBM Deutschland Informationssysteme GmbH 2002

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
Mai 2002

Inhaltsverzeichnis

Abbildungsverzeichnis	v	Kapitel 4. Erste Schritte mit WebSphere Studio Application Developer	19
Kapitel 1. Zu diesem Handbuch	1	Dauer	19
Übersicht zu diesem Lernprogramm	1	Übersicht	19
Vorausgesetzte Kenntnisse	1	Voraussetzungen	19
Dauer des Lernprogramms	2	Video Central-Tasks	19
Vorbereitungen	2	Task 1: Definieren des WAS-Exemplars für den fernen Server	19
In diesem Lernprogramm verwendete Konventionen	2	Task 2: Definieren des Webprojekts Video-Central und Importieren des Quellcodes	24
Kapitel 2. Beispielanwendung von Video Central	3	Zusammenfassung der Lerneinheit	30
Dauer	3	Nächster Schritt	30
Vorbereitungen	3	Kapitel 5. Prüfen des Video Central-Designs	31
Übersicht	3	Dauer	31
Web-Services - Einführung	3	Übersicht	31
Was ist ein Web-Service?	3	Video Central-Design	31
Zugreifen auf Web-Services	5	Webschnittstellenschicht (WIL)	31
Szenario von Video Central	6	Entwickeln der Unternehmenslogikschicht (BLL)	31
Grundlegende Unternehmensfaktoren	6	Erstellen von Datenmodellen	42
Video Central-Architektur	8	Zusammenfassung der Lerneinheit	45
Zusammenfassung der Lerneinheit	11	Nächster Schritt	45
Nächster Schritt	11	Kapitel 6. Implementieren von Video Central	47
Kapitel 3. Aktivieren der DB2-Umgebung für Video Central	13	Dauer	47
Dauer	13	Voraussetzungen	47
Vorbereitungen	13	Übersicht	47
Systemvoraussetzungen	13	Video Central-Tasks	47
Übersicht	13	Task 1: Vorbereiten der Umgebung für das Implementieren von Video Central vom WebSphere Studio Application Developer	47
Video Central-Tasks	13	Task 2: Implementieren der Web-Services im WebSphere Application Server	49
Task 1: Erstellen der Benutzer-ID 'ivc'	13	Zusammenfassung der Lerneinheit	57
Task 2: Erstellen der Beispieldatenbank	14	Nächster Schritt	57
Task 3: Aktivieren der Beispieldatenbank für die Verwendung von DB2 XML Extender	14	Kapitel 7. Erstellen und Ausführen eines webbasierten Beispielclients für Video Central	59
Task 4: Aktivieren der Beispieldatenbank für die Verwendung von DB2 Net Search Extender	16	Dauer	59
Task 5: Erstellen und Füllen der Video Central-Tabellen	16	Voraussetzungen	59
Zusammenfassung der Lerneinheit	17	Übersicht	59
Nächster Schritt	17		

Design des Beispielclients von Video Central	60	DB2 XML Extender - Produktbeschreibung.	81
Klassen- und Sequenzdiagramme für den		DB2 Net Search Extender - Produkt-	
Beispielclient von Video Central	61	beschreibung.	82
Video Central-Tasks	63	IBM WebSphere Studio Application Developer	
Task 1: Vorbereiten des Clientprojekts . .	63	- Produktbeschreibung	83
Task 2: Erstellen der Proxies mit		Informationen zur Web-Services-Umge-	
WebSphere Studio Application Developer .	71	bung	84
Task 3: Implementieren des Beispielclients		Informationen zur XML-Entwicklungs-	
im WebSphere Application Server	74	umgebung	85
Task 4: Ausführen des Beispielclients. .	75		
Zusammenfassung der Lerneinheit	75	Anhang B. Glossar	89
		Anhang C. Bemerkungen	91
Kapitel 8. Beheben allgemeiner Fehler . .	77	Kontaktaufnahme mit IBM.	93
Kapitel 9. Zusätzliche Informationen . . .	79	Produktinformationen.	93
Anhang A. Referenzinformationen	81		

Abbildungsverzeichnis

1. Grundlegende Aufgabenbereiche von Web-Services	4	21. Ausführung des WAS Admin Server prüfen	49
2. Video Central-Architektur	10	22. Auswahl des zu implementierenden Web-Services	50
3. Erstellen und Konfigurieren eines neuen Serverexemplars	20	23. Zu implementierende Java-Bean	51
4. Erstelltes Serverexemplar	21	24. Konfigurieren der Java-Bean als Web-Service	52
5. Konfigurieren der Datenquelle	23	25. Auswahl der zu implementierenden öffentlichen Methoden.	53
6. Hinzufügen externer JAR-Dateien	25	26. Web Service Binding Proxy Generation	54
7. Hinzufügen des Webprojekts zum Server	26	27. Inhalt der WSDL-Datei	55
8. Hinzufügen des Webprojekts zum Server (Fortsetzung)	27	28. Alle Web-Services implementiert	56
9. Vor dem Import der JAR-Dateien	28	29. Grundlegende Aufgabenbereiche von Web-Services	60
10. Auswahl der zu importierenden Ressourcen.	29	30. Klassendiagramm für den Beispielclient Video Client von Video Central.	61
11. Nach der Vorbereitung des Video Central-Projekts	30	31. Sequenzdiagramm für den Beispielclient Video Client von Video Central.	62
12. Video Central - Übersichtsklassendiagramm	32	32. Hinzufügen externer JAR-Dateien	64
13. Klassendiagramm zum Web-Service für Kundenverstöße	34	33. Hinzufügen des Client-Web-Projekts zum Server	65
14. Klassendiagramm zum Web-Service für Wunschnlisten	36	34. Hervorheben von 'source'	66
15. Sequenzdiagramm zum Hinzufügen von Kundenverstößen	38	35. Importieren des Client-Codes von einer JAR-Datei in das Projekt VideoClient	67
16. Sequenzdiagramm zur Abfrage von Kundenverstößen	40	36. Hervorheben von 'webApplication'	68
17. Sequenzdiagramm zum Hinzufügen einer Wunschliste	41	37. Anzeige 'File system'	69
18. Diagramm des Datenmodells - Teil 1	43	38. Erfolgreicher Import des Video Central-Clients VideoClient.	70
19. Diagramm des Datenmodells - Teil 2	44	39. Generieren des BusinessRegistration-Proxies	72
20. Prüfen des HTTP-Servers.	48	40. Für alle WSDL-Dateien erstellte Proxies	73
		41. Hinzufügen der Client-Servlets	74

Kapitel 1. Zu diesem Handbuch

Übersicht zu diesem Lernprogramm

Das IBM Video Central-Lernprogramm umfasst in einzelne Arbeitsschritte untergliederte Anweisungen und eine vollständige Übungsanwendung, mit deren Hilfe der Einsatz von DB2 als Web-Service-Provider veranschaulicht wird. Web-Services basieren auf den neuesten Technologien und umfassen z. B. die Spezifikationen von SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) und UDDI (Universal Description, Discover and Integration). Bei SOAP handelt es sich um ein Standardnachrichtenformat, das zum Aufrufen von Web-Services verwendet wird. Mit der Anwendung des Video Central-Lernprogramms können SOAP-Nachrichten mit Hilfe des Internet-Standardprotokolls HTTP (Hyper Text Transfer Protocol) übertragen werden. Web-Service-Requester können mit Hilfe des WSDL-Dokuments des Web-Service-Providers entwickelt werden. Weitere Informationen zu Web-Services finden Sie unter der Adresse <http://www.ibm.com/software/webservices>.

Bei den Web-Services handelt es sich um ein neues Entwicklungsmodell für die Integration von Anwendungen. Es bietet dem Benutzer ein Standardverfahren zum Beschreiben, Veröffentlichen, Lokalisieren und Aufrufen von Unternehmenslogikkomponenten innerhalb einer verteilten Datenverarbeitungsumgebung. Das vorliegende Lernprogramm umfasst die vollständige Dokumentation, die Sie zum Implementieren von sechs Web-Services auf einem WebSphere Application Server (WAS) benötigen. Darüber hinaus wird auch eine Clientanwendung bereitgestellt, mit der Sie diese Web-Services unter realen Arbeitsbedingungen ausführen können. Im vorliegenden Lernprogramm werden die dynamischen XML-Zuordnungsfunktionen von DB2 XML Extender und die Hochgeschwindigkeitsengine für die Volltextsuche, die in DB2 Net Search Extender implementiert wurde, erläutert.

Vorausgesetzte Kenntnisse

Dieses Lernprogramm richtet sich an den professionellen Java-Programmierer, der über umfassende Kenntnisse in folgenden Bereichen verfügt:

- WebSphere Application Server
- Web-Service-Konzepte (weitere Informationen hierzu enthält "Kapitel 2. Aktivieren der DB2-Umgebung für Video Central")
- DB2 Universal Database
- Konzepte der objektorientierten Programmierung
- Programmiersprache Java

Dauer des Lernprogramms

Zum Durcharbeiten des vorliegenden Lernprogramms werden ca. 2 Stunden benötigt.

Vorbereitungen

Die folgenden Produkte werden im Lernprogramm benutzt und müssen deswegen vorab auf dem verwendeten System installiert werden:

- WebSphere Application Server (WAS), Version 4.0 Advanced Edition Single Server
- DB2 Universal Developer's Edition (DB2 UDE), Version 8.1 einschließlich DB2 XML Extender und DB2 Net Search Extender (DB2 XML Extender und DB2 Net Search Extender sind im Lieferumfang enthalten, befinden sich jedoch auf einer eigenen CD und müssen separat installiert werden).
- WebSphere Studio Application Developer (WSAD), Version 4.0

In diesem Lernprogramm verwendete Konventionen

Im vorliegenden Lernprogramm gilt Folgendes:

- <DB2_install_dir> wird verwendet, um auf das Installationsverzeichnis von DB2 zu verweisen.
- <WAS_install_dir> wird verwendet, um auf das Installationsverzeichnis von WebSphere Application Server zu verweisen.
- <WSAD_install_dir> wird verwendet, um auf das Installationsverzeichnis von WebSphere Studio Application Developer zu verweisen.

Kapitel 2. Beispielanwendung von Video Central

Dauer

Zum Durcharbeiten des vorliegenden Kapitels benötigen Sie ca. 20 Minuten.

Vorbereitungen

Sie sollten mit den Konzepten der objektorientierten Programmierung vertraut sein.

Übersicht

Im vorliegenden Kapitel werden Sie Informationen zu den verfügbaren Web-Services und zu SOAP (Simple Object Access Protocol) erhalten. Darüber hinaus lernen Sie die in Video Central implementierten Web-Services sowie die Video Central-Architektur kennen.

Web-Services - Einführung

Im folgenden Abschnitt finden Sie eine Übersicht zu den verfügbaren Web-Services.

Was ist ein Web-Service?

Bei einem Web-Service handelt es sich um eine Gruppe zusammengehöriger Anwendungsfunktionen, die von einer bestimmten Anwendung aufgerufen werden können. Anders als bei herkömmlichen Webanwendungen, die zur Informationsdarstellung HTML verwenden, wird bei Web-Services zum Informationsaustausch XML eingesetzt. Die Implementierung und der Zugriff auf Web-Services kann über zahlreiche unterschiedliche Programmiersprachen und Plattformen erfolgen. Im vorliegenden Lernprogramm wird die Programmiersprache Java verwendet, um den Web-Service-Requester und -Provider auf einer Windows-Plattform zu implementieren.

Bei Web-Services handelt es sich um modulare Anwendungen, die zur Ausführung spezifischer Tasks dienen und auf die über offene Protokolle wie z. B. HTTP zugegriffen werden kann. Im Unternehmen können diese Web-Services dynamisch kombiniert und abgestimmt werden, so dass sie zur Ausführung komplexer Operationen auf der Basis eines flexiblen Programmierungsmodells eingesetzt werden können.

Das Konzept der Web-Services basiert auf der Definition von drei grundlegenden Aufgabenbereichen (vgl. Abbildung 1).

Service-Provider

- Bereitstellung von e-business-Services
- Veröffentlichung der Verfügbarkeit dieser Services über eine Registrierdatenbank (optional)

Service-Broker

- Unterstützung von Veröffentlichungs- und Lokalisierungsservices

Service-Requester

- Lokalisieren erforderlicher Services mit Hilfe des Service-Brokers
- Herstellung von Verbindungen zu den vom Service-Provider bereitgestellten Services
- Aufrufen der Services über eine definierte Schnittstelle

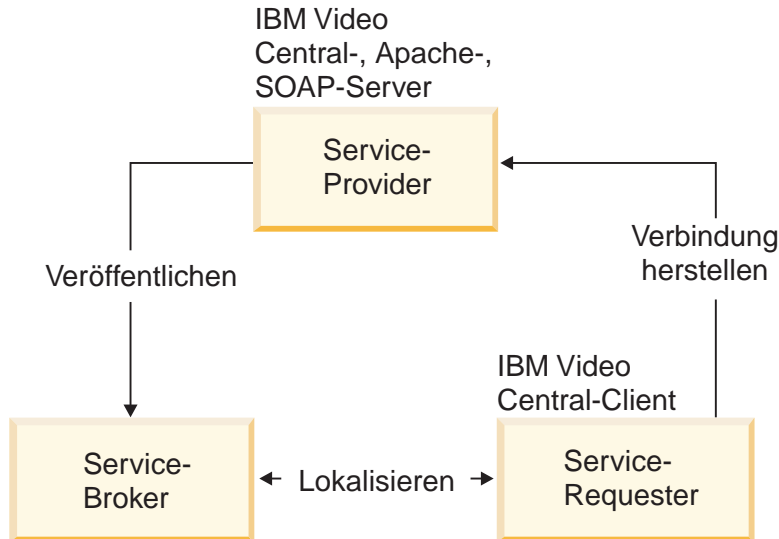


Abbildung 1. Grundlegende Aufgabenbereiche von Web-Services

Die zentrale Anlaufstelle für Web-Services, die von Unternehmen zur Bildung von Partnerschaften verwendet wird, wird als UDDI-Registrierdatenbank (UDDI = Universal Description, Discovery and Integration) bezeichnet. In der UDDI-Spezifikation werden offene, plattformunabhängige Standards definiert, mit deren Hilfe Unternehmen Informationen in einer globalen Unternehmensregistrierdatenbank gemeinsam verwenden, die Services der anderen Teilnehmer prüfen und die Art und Weise der Kommunikation über das Internet definieren können. Diese Registrierdatenbank soll zu einer umfangreichen Bibliothek mit Funktionen ausgebaut werden, die von Entwicklern zum schnellen Erstellen dynamischer Anwendungen eingesetzt werden können.

IBM nimmt bei der Definition der UDDI-Spezifikation, die bereits von zahlreichen führenden Wirtschaftsunternehmen übernommen wurde, eine Schlüsselposition ein.

Web-Services werden in der UDDI-Unternehmensregistrierdatenbank mit Hilfe von WSDL (Web Services Description Language) veröffentlicht. Der Informationsaustausch zwischen diesen Web-Services wird über SOAP (Simple Open Access Protocol) realisiert. Web-Services können in die drei folgenden Kategorien unterteilt werden:

- *Unternehmensinformationsservices* - Hierbei stellen Unternehmen Informationen für die gemeinsame Benutzung durch Konsumenten und andere Unternehmen bereit. In diesem Fall werden die Web-Services vom Unternehmen eingesetzt, um seinen Wirkungsbereich auszudehnen. Beispiele für Unternehmensinformationsservices sind Nachrichtenservices sowie Services für Wettermeldungen und Börsennotierungen.
- *Unternehmensintegrationsservices* - Hierbei stellen Unternehmen ihren Kunden gebührenpflichtige Services zur Ausführung bestimmter Dienstleistungen zur Verfügung. In diesem Fall baut das Unternehmen ein globales Netzwerk von Lieferanten auf, das zur Durchführung der Geschäftstransaktionen verwendet werden kann. Beispiele für Unternehmensintegrationsservices sind Gebots- und Auktions-E-Marketplaces, Reservierungssysteme und Services zur Prüfung von Krediten.
- *Services für die Unternehmensprozessauslagerung* - Hierbei setzt sich ein Unternehmen von der Konkurrenz dadurch ab, dass es eine globale Wertekette aufbaut. In diesem Fall verwendet das Unternehmen Web-Services, um seine Prozesse dynamisch zu integrieren.

Weitere Informationen zu UDDI finden Sie unter der Adresse <http://www.uddi.org>

Zugreifen auf Web-Services

Ein Unternehmen kann die Unternehmensfunktionen, die im Web angeboten werden, veröffentlichen. Über entsprechende Registrierdatenbanken wie z. B. die UDDI-Unternehmensregistrierdatenbank können potenzielle Konsumenten und Kunden feststellen, dass diese Services im Web angeboten werden.

Web-Services bilden eine natürliche Erweiterung der Modelle für verteilte oder Client/Server-Anwendungen. Bei Web-Services können Methoden für ferne Objekte mit Hilfe eines unter der Bezeichnung SOAP bekannten XML-Standardformats aufgerufen werden. Die SOAP-Nachricht kann über verschiedene Übertragungsprotokolle einschließlich MQSeries und HTTP übertragen werden. Die UDDI-Registrierdatenbank stellt ein Verzeichnis aller verfügbaren Services und Service-Provider bereit. Die Schnittstelle basiert auf dem sprachenunabhängigen XML-Standard WSDL.

Die Schlüsselprodukte von IBM unterstützen Web-Service-Funktionen. Darüber hinaus stehen Tools zum automatischen Generieren, Veröffentlichen, Lokalisieren, Testen und Aufrufen von Web-Services zur Verfügung. Das vorliegende Lernprogramm erläutert die Verwendung verschiedener dieser Tools. IBM stellt außerdem eine weitere Möglichkeit zur Übergabe von SQL-Anweisungen und bei Bedarf zur Steuerung des Formats zur Verfügung, das bei der Rückgabe von Daten verwendet wird. Diese Unterstützungsfunktion wird als WOLF (Web Services Object Runtime Framework) und DADX (Document Access Definition eXtension) bezeichnet. Weitere Informationen hierzu finden Sie unter folgender Adresse:

<http://www.ibm.com/software/data/pubs/papers/db2webservices/db2webservices.pdf>

Szenario von Video Central

Bei Video Central handelt es sich um einen generischen Web-Service-Provider für Videoverleihanwendungen (B2B-Anwendungen). Video Central dient zur Bereitstellung eines zentralen Datenrepositorys, auf das über registrierte Webanwendungen zugegriffen werden kann. Die Video Central-Anwendung umfasst eine Gruppe von Web-Services, um die Unternehmensanforderungen der Kunden in zwei Bereichen abzudecken: Hierbei dienen die **Unternehmensservices** zur Erfüllung der administrativen Anforderungen der Kunden und die **Kundenservices** zur Bereitstellung von Funktionen, mit denen die Kunden ihren eigenen Kundenstamm bedienen und verwalten können.

Grundlegende Unternehmensfaktoren

Die Videoverleihunternehmen müssen beim täglichen Verleihgeschäft die folgenden Rahmenbedingungen und Einschränkungen berücksichtigen:

- Manuelle und wiederholte Dateneingabe für neu in den Verleihbetrieb aufgenommene Titel.
- Eingeschränkte Titelbeschreibungen (manuelle Eingabe von Informationen zur Handlung, zu den Rollen und zur Besetzung).
- Nichtvorhandensein einer Funktion, mit der den Kunden auf der Basis Ihrer Leihhistorie und der bevorzugten Filmgenres Vorschläge zu potenziell interessanten Titeln gemacht werden können.
- Bestimmte Zusatzfunktionen (z. B. Wunschlisten oder Historie der zuvor ausgeliehenen Titel) sind auf die lokal verfügbaren Daten begrenzt.
- Eingeschränkte Möglichkeit der Auswertung von Kundendaten für neue und wiederkehrende Kunden zur Beurteilung des Risikos beim Verleih teurer Verleihartikel.
- Verluste auf Grund von Titeln, die von Neukunden unterschlagen (geliehen und nicht zurückgebracht) wurden.

Zusammenfassend kann festgehalten werden, dass die Verleihunternehmen nur auf die Informationen zugreifen können, die in der jeweils eigenen Datenbank gespeichert sind. In der aktuellen Phase stellt Video Central die im Folgenden aufgeführten Services bereit.

Video Central-Unternehmensservices

Für das vorliegende Lernprogramm wurde als unternehmensbezogener Web-Service ein Service für die Unternehmensregistrierung entworfen.

Unternehmensregistrierung: Dieser Web-Service ermöglicht einem Kunden (z. B. einer Videothek) die Registrierung seines Unternehmens bei Video Central. Nach der Registrierung bei Video Central kann das Unternehmen die anderen verfügbaren Web-Services nutzen.

Video Central-Kundenservices

Im vorliegenden Fall wurden fünf spezifische kundenbezogene Web-Services entworfen: Kundenregistrierung, Protokollierung von Kundenverstößen gegen die Verleihbedingungen, Erstellung von Listen der entliehenen Titel, Erstellung der Wunschliste des Kunden und Suche nach bestimmten Filmtiteln.

Kundenregistrierung: Dieser Web-Service ermöglicht registrierten Unternehmen (d. h. Videotheken) die Registrierung ihrer Kunden bei Video Central. Nach der Registrierung des Kunden können wichtige Kundendaten (z. B. Leihhistorie, Wunschliste, Verstöße gegen die Verleihbedingungen etc.) gespeichert werden. Mit Hilfe der sonstigen verfügbaren Services können diese Daten anschließend analysiert und abgerufen werden.

Kundenverstöße gegen die Verleihbedingungen: Dieser Web-Service ermöglicht registrierten Unternehmen, Informationen zu Verstößen eines Kunden gegen die Verleihbedingungen zum Video Central-Repository hinzuzufügen. Hierdurch kann das Unternehmen derartige Verstöße protokollieren, ohne dass die entsprechenden Daten in der lokalen Datenbank gespeichert werden müssen. Dieser Web-Service ermöglicht registrierten Unternehmen darüber hinaus, eventuell vorhandene Verstöße eines potenziellen oder wiederkehrenden Kunden jederzeit abzufragen. Da es sich bei Video Central um ein zentrales Repository für eine größere Anzahl von Videotheken handelt, kann das Unternehmen vor dem Verleihen eines neuen Artikels auch feststellen, ob ein Kunde bereits gegen die Geschäftsbedingungen einer anderen Videothek verstoßen hat.

Liste der entliehenen Titel: Dieser Web-Service ermöglicht dem registrierten Unternehmen, für jeden Kunden eine Liste der in der Vergangenheit entliehenen Videos zu erstellen. Darüber hinaus kann das registrierte Unternehmen die Leihvorgänge eines bestimmten Kunden jederzeit abfragen.

Wunschliste des Kunden: Dieser Web-Service ermöglicht dem registrierten Unternehmen die Erfassung der Filmtitel, die der Kunde als interessant angegeben hat. Hierdurch kann das Unternehmen diese Titel für den Kunden protokollieren und die Wunschliste jederzeit abfragen. Darüber hinaus kann die Videothek diese Liste verwenden, um festzustellen, welche Videos von den Kunden nachgefragt werden und ihren Bestand an Verleihtiteln entsprechend anpassen.

Suche nach bestimmten Filmtiteln: Dieser Web-Service ermöglicht registrierten Unternehmen die Suche nach Filmtiteln und den Zusammenfassungen von Filmhandlungen. Das Unternehmen muss die Daten nun nicht mehr manuell in eine lokale Datenbank eingeben, sondern kann online nach aktuellen und kontinuierlich aktualisierten Filminformationen suchen. Dieser Web-Service kann von Videothekenmitarbeitern oder -kunden zur Entscheidungsfindung verwendet werden.

Video Central-Architektur

Die Video Central-Anwendung basiert auf einer Web-Service-Architektur. Weitere Einzelheiten zum Design von Video Central finden Sie in "Kapitel 5. Prüfen des Video Central-Designs". Entsprechend dieser Architektur stellt Video Central primär einen dynamischen und flexiblen Provider für e-business-Services (über das Internet) dar.

Es können entsprechende Service-Requester für die Verwendung der von Video Central bereitgestellten Services entwickelt werden. Wenn ein Service-Requester einen bestimmten Service von Video Central anfordert, wird diese Anforderung an das Service-Proxy-Objekt auf der Clientmaschine gesendet. Der Proxy packt die Anforderung und die zugehörigen Daten in einem XML-Format und versendet diese XML-Daten in einer SOAP-Nachricht. Das SOAP-Laufzeitervlet wird in WAS Version 4.x bereitgestellt. Nach dem Empfang der Anforderung durch das SOAP-Laufzeitervlet wird diese an die entsprechende Java-Bean delegiert, die den geeigneten Service zur Verarbeitung der Anforderung implementiert.

Die Video Central-Anwendung ist in den folgenden drei Schichten implementiert:

1. Webschnittstellenschicht (Web Interface Layer = WIL)
2. Unternehmenslogikschicht (BLL = Business Logic Layer)
3. Datenzugriffsschicht (Data Access Layer = DAL)

Dieses dreischichtige Design ermöglicht eine flexible komponentenbasierte Implementierung von Video Central und die einfache Erweiterung der Anwendung über eine erweiterbare Rahmenanwendung.

Die Webschnittstellenschicht (WIL = Web Interface Layer) kann implementiert werden, um unterschiedliche Webschnittstellen wie z. B. Web-Services, HTTP-Servlet oder JavaServer Pages (JSP) etc. zu unterstützen. Hierdurch ergibt sich eine optimale Flexibilität in Bezug auf das Design der Endbenutzerschnittstelle. Video Central implementiert Web-Services mit Hilfe der SOAP-Laufzeitkomponente für den Aufruf von Web-Services.

Die Unternehmenslogikschicht (BLL = Business Logic Layer) wurde mit Hilfe von Web-Services als Abstraktionsebene entworfen. Jeder Web-Service definiert eine Gruppe von Unternehmenslogikkomponenten und die Schnittstelle wird mit Hilfe von WSDL (Web Services Description Language) beschrieben. Die WSDL-Schnittstelle kann ihrerseits in einer öffentlichen oder privaten UDDI-Registrierdatenbank (UDDI = Universal Description, Discovery, and Integration) veröffentlicht werden. Eine Schlüsselkomponente der Video Central-Anwendung basiert auf dem Einsatz von DB2 XML Extender zur Verwaltung des Repositorys für die Kundenverträge, um Daten speichern und abrufen zu können. Darüber hinaus verwendet Video Central DB2 Net Search Extender, um seinen Web-Service für die Filmtitelsuche über seine schnelle Textsuchengine zu ermöglichen.

Die Datenzugriffsschicht (Data Access Layer = DAL) wurde mit Hilfe der IBM Data Access-Beans (Oberfläche für Java Database Connectivity - JDBC) innerhalb der Data Access-Komponenten konsolidiert. Die JavaBeans, die zur Implementierung der Unternehmenslogik dienen, verwenden die Data Access-Komponenten zum Speichern und Abrufen von Daten. DB2 UDB wird als Datenbankserver für die Video Central-Anwendung verwendet.

Abbildung 2 zeigt die Architektur der Video Central-Anwendung.

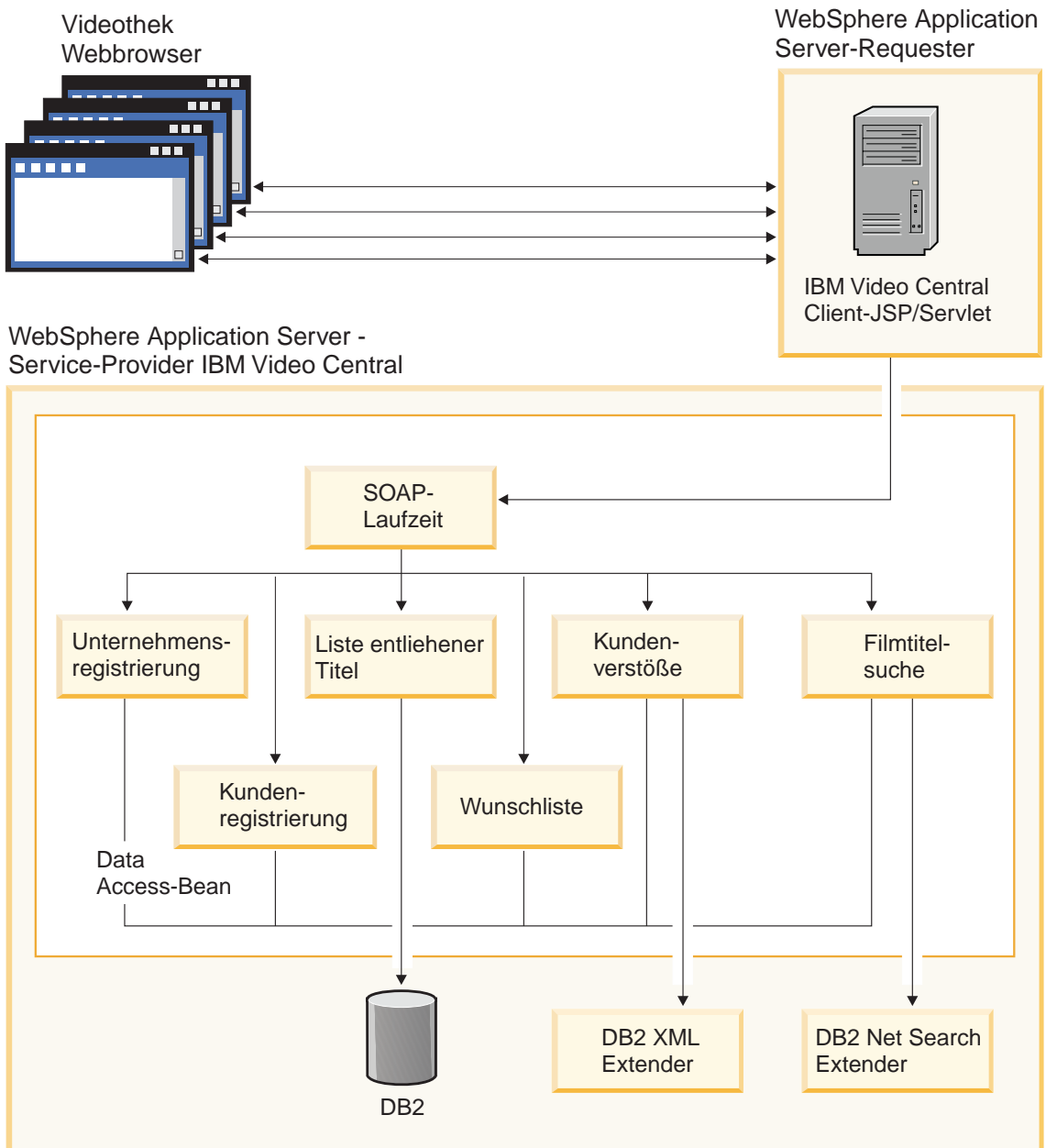


Abbildung 2. Video Central-Architektur

Zusammenfassung der Lerneinheit

In diesem Kapitel wurden die folgenden Themen erläutert:

- Web-Services
- Grundlagen der übergeordneten Architektur und des Designs der Video Central-Anwendung

Nächster Schritt

Als Nächstes werden Sie lernen, welche DB2-Entwicklungsumgebung erforderlich ist und wie diese Umgebung für die Implementierung der Video Central-Anwendung vorbereitet werden muss.

Kapitel 3. Aktivieren der DB2-Umgebung für Video Central

Dauer

Zum Durcharbeiten des vorliegenden Kapitels benötigen Sie ca. 15 Minuten.

Vorbereitungen

Systemvoraussetzungen

- Microsoft Windows NT Version 4.0, Service Pack 6a oder höher bzw. Windows 2000
 - Java 2 SDK Standard Edition Version 1.3
 - DB2 Universal Developer's Edition (DB2 UDE) Version 8.1
 - DB2 XML Extender (im Lieferumfang von DB2 UDE Version 8.1 enthalten, jedoch auf eigener CD gespeichert und separat zu installieren)
 - DB2 Net Search Extender (im Lieferumfang von DB2 UDE Version 8.1 enthalten, jedoch auf eigener CD gespeichert und separat zu installieren)
-

Übersicht

Der vorliegende Abschnitt erläutert die Arbeitsschritte, die zur Konfiguration der DB2-Umgebung für Video Central erforderlich sind. Dieser Prozess umfasst Folgendes:

- Aktivieren der Video Central-Datenbank für DB2 XML Extender und DB2 Net Search Extender

Darüber hinaus müssen Sie folgende Arbeitsschritte ausführen:

- Erstellen eines neuen Benutzers mit dem Namen `ivc`
 - Erstellen der für Video Central erforderlichen Tabellen
-

Video Central-Tasks

Task 1: Erstellen der Benutzer-ID 'ivc'

Wenn dieser Arbeitsschritt noch nicht ausgeführt wurde, müssen Sie nun im verwendeten Betriebssystem die Benutzer-ID `ivc` erstellen.

Windows 2000:

1. Klicken Sie auf **Start** → **Einstellungen** → **Systemsteuerung**.
2. Wählen Sie **Verwaltung** → **Computerverwaltung** → **Lokale Benutzer und Gruppen** aus.

3. Klicken Sie mit der rechten Maustaste in der Menüleiste auf **Benutzer** —> **Neuer Benutzer**.
4. Erstellen Sie einen neuen Benutzer mit dem Namen `ivc` und dem Kennwort `VideoCentral`. Nehmen Sie die Auswahl von **Benutzer muss Kennwort bei der nächsten Anmeldung ändern** zurück.

Windows NT:

1. Klicken Sie auf **Start** —> **Verwaltung (Allgemein)** —> **Benutzer-Manager**.
2. Klicken Sie mit der rechten Maustaste in der Menüleiste auf **Benutzer** —> **Neuer Benutzer**.
3. Erstellen Sie einen neuen Benutzer mit dem Namen `ivc` und dem Kennwort `VideoCentral`. Nehmen Sie die Auswahl von **Benutzer muss Kennwort bei der nächsten Anmeldung ändern** zurück.

Task 2: Erstellen der Beispieldatenbank

Im vorliegenden Lernprogramm wird als Video Central-Datenbank die Datenbank `sample` verwendet. Zum Erstellen der Datenbank `sample` müssen Sie einen der folgenden Arbeitsschritte ausführen:

- Führen Sie in der Befehlszeile den Befehl `db2samp1` aus.
- Klicken Sie auf **Start** —> **Programme** —> **IBM DB2** —> **Erste Schritte**. Klicken Sie anschließend im Fenster **Erste Schritte** auf **Beispieldatenbanken erstellen**.

Task 3: Aktivieren der Beispieldatenbank für die Verwendung von DB2 XML Extender

Zum Speichern oder Abrufen von XML-Dokumenten aus DB2 mit Hilfe von DB2 XML Extender müssen Sie zuerst die Datenbank für den Einsatz von XML aktivieren.

Wenn eine Datenbank für XML aktiviert wird, führt DB2 XML Extender die folgenden Operationen aus:

- Erstellen aller benutzerdefinierten Typen (UDTs = User-Defined Types) und benutzerdefinierten Funktionen (UDFs = User-Defined Functions)
- Erstellen und Füllen von Steuertabellen mit den für DB2 XML Extender erforderlichen Metadaten
- Erstellen des Schemas `db2xml` und Zuordnen der erforderlichen Zugriffsrechte

Im Folgenden sind die Arbeitsschritte aufgeführt, die zum Aktivieren der Datenbank `sample` für den Einsatz von DB2 XML Extender ausgeführt werden müssen.

1. Führen Sie im DB2-Befehlsfenster den Befehl
`<DB2_install_dir>\tutorials\VideoCentral\db\dbEnableXML.bat` aus.

Dieses Script führt die folgenden Operationen aus:

- a. Herstellen der Verbindung zur Datenbank `sample`.
- b. Binden der XML Extender-Bindedateien an die Datenbank `sample`.
- c. Aktivieren von DB2 XML Extender.

Anmerkung: Das Script `<DB2_install_dir>\tutorials\VideoCentral\db\dbDisableXML.bat` wird für den Fall zur Verfügung gestellt, dass die XML-Unterstützung für die Datenbank inaktiviert werden muss. Im Rahmen des vorliegenden Lernprogramms ist dieser Arbeitsschritt nicht erforderlich.

2. Prüfen Sie, ob DB2 XML Extender aktiv ist. Geben Sie hierzu in einem DB2-Befehlsfenster die folgenden Befehle ein:
 - a. `db2 connect to sample.`
 - b. `db2 select procname from syscat.procedures where procschema='DB2XML'.`

Daraufhin wird eine Liste mit den folgenden Namen gespeicherter Prozeduren angezeigt:

- DXXDISABLECOLL
- DXXDISABLECOLUMN
- DXXENABLECOLL
- DXXENABLECOLUMN
- DXXGENXML
- DXXGENXMLCLOB
- DXXINSERTXML
- DXXRETRIEVEXML
- DXXRETRIEVEXMLCLOB
- DXXSHREDXML

Die Video Central-Anwendung verwendet die gespeicherten Prozeduren `DXXGENXML` und `DXXSHREDXML`, die beide für den Web-Service zur Protokollierung von Kundenverstößen verwendet werden. `DXXGENXML` wird zum Abfragen von Kundenverstößen verwendet. Die entsprechenden Daten werden von DB2 im XML-Dokumentformat abgerufen. `DXXSHREDXML` wird beim Hinzufügen von Daten zu einem Kundenverstoß verwendet, wobei die empfangenen Daten ebenfalls als XML-Dokumente in DB2 importiert werden.

Zusätzliche Informationen zu DB2 XML Extender finden Sie im Handbuch *XML Extender Verwaltung und Programmierung*, das online unter der Adresse http://www.ibm.com/software/data/db2/extenders/xml_ext/library.html zur Verfügung steht.

Task 4: Aktivieren der Beispieldatenbank für die Verwendung von DB2 Net Search Extender

1. Führen Sie im DB2-Befehlsfenster den Befehl
<DB2_install_dir>\tutorials\VideoCentral\db\dbEnableNX.bat aus.
Dieses Script führt die folgenden Operationen aus:
 - a. Starten von DB2.
 - b. Herstellen der Verbindung zur Datenbank sample.
 - c. Starten des Dämons von DB2 Net Search Extender.
 - d. Aktivieren der Datenbank für DB2 Net Search Extender.
 - e. Aktivieren der Textspalten (für Video Central wird die Netzwerksuche auf die folgenden Spalten indexiert: ivc.titleinfo.name und ivc.titleplot-plotsummary).

Anmerkung: Das Script

<DB2_install_dir>\tutorials\VideoCentral\db\dbDisableNX.bat wird für den Fall zur Verfügung gestellt, dass die Verwendung von DB2 Net Search Extender für die Datenbank inaktiviert werden soll. Im Rahmen des vorliegenden Lernprogramms ist dieser Arbeitsschritt nicht erforderlich.

2. Prüfen Sie, ob DB2 Net Search Extender aktiv ist. Öffnen Sie hierzu die Datei <DB2_install_dir>\tutorials\VideoCentral\db\movieSearch.log, und stellen Sie fest, ob Fehler aufgelistet werden.

Anmerkung: Sind Fehler vorhanden, finden Sie weitere Informationen in "Kapitel 8. Beheben allgemeiner Fehler".

Zusätzliche Informationen zu DB2 Net Search Extender finden Sie im Handbuch *Net Search Extender Verwaltung und Programmierung*, das online unter der Adresse <http://www.ibm.com/software/data/db2/extenders/netsearch> zur Verfügung steht, sowie im "Anhang A - Referenzinformationen".

Task 5: Erstellen und Füllen der Video Central-Tabellen

1. Öffnen Sie ein DB2-Befehlsfenster, um die Stapeldatei auszuführen.
2. Führen Sie im DB2-Befehlsfenster den Befehl
<DB2_install_path>\tutorials\VideoCentral\db\tblCreate.bat aus.
Dieses Script führt die folgenden Operationen aus:
 - a. Erstellen der für Video Central erforderlichen Tabellen in der Datenbank sample.
 - b. Erstellen der erforderlichen Indizes und Integritätsbedingungen.
 - c. Füllen der erforderlichen Tabellen (Codetabellen, DAD-Tabelle, Tabellen für die Filmtitelsuche).

Anmerkung: Das Script <DB2_install_dir>\tutorials
\VideoCentral\db\tblClean.bat wird für den Fall zur Verfügung gestellt, dass Video Central nicht mehr ausgeführt werden soll und die Video Central-Tabellen gelöscht werden sollen. Im Rahmen des vorliegenden Lernprogramms ist dieser Arbeitsschritt nicht erforderlich.

3. Geben Sie in einem DB2-Befehlsfenster die folgenden Befehle ein:
 - a. DB2 connect to sample.
 - b. DB2 list tables for schema ivc.

Daraufhin wird eine Liste den folgenden 10 Tabellen angezeigt:

- BUSINESSINFO
- DAD
- QUERY_RESULT_TAB
- TITLEINFO
- TITLEPLOT
- USERINFO
- USERINFRACT
- USERINFRACT_TMP
- USERRENTALS
- USERWISHES

Zusammenfassung der Lerneinheit

In diesem Kapitel wurden die folgenden Themen erläutert:

- Erstellen eines neuen Benutzers mit dem Namen ivc
- Erstellen der Datenbank sample
- Aktivieren der Video Central-Datenbank für DB2 XML Extender und DB2 Net Search Extender
- Erstellen und Füllen der für Video Central erforderlichen Tabellen

Nächster Schritt

Als Nächstes werden Sie lernen, welche Entwicklungsumgebung erforderlich ist und wie diese Umgebung für die Ausführung von Video Central vorbereitet werden muss.

Kapitel 4. Erste Schritte mit WebSphere Studio Application Developer

Dauer

Zum Durcharbeiten des vorliegenden Kapitels benötigen Sie ca. 20 Minuten.

Übersicht

Dieses Kapitel gibt eine kurze Einführung in IBM WebSphere Studio Application Developer. Es erläutert, wie die Java-Entwicklungsumgebung für den Einsatz von Web-Services konfiguriert werden kann.

Voraussetzungen

- WebSphere Application Server (WAS) Version 4.0, Advanced Edition Single Server
 - WebSphere Studio Application Developer (WSAD) Version 4.0
-

Video Central-Tasks

Task 1: Definieren des WAS-Exemplars für den fernen Server

Diese Task besteht im Erstellen eines Video Central-Serverprojekts und im Erstellen der Definition für den fernen WAS-Server innerhalb von WebSphere Studio Application Developer.

Starten von WebSphere Studio Application Developer

Starten Sie WebSphere Studio Application Developer, indem Sie auf **Start** → **Programme** → **IBM WebSphere Studio Application Developer** → **IBM WebSphere Studio Application Developer** klicken.

Erstellen und Konfigurieren des Video Central-Servers

1. Erstellen Sie den Video Central-Server:
 - a. Erstellen Sie in der Serverperspektive ein Serverprojekt, indem Sie auf **File** → **New** → **Server Project** klicken.
 - b. Ordnen Sie dem Projekt den Namen VideoCentralServer zu. Klicken Sie auf **Finish**.
2. Erstellen Sie ein neues Serverexemplar und konfigurieren Sie dieses:
 - a. Klicken Sie in der Serverperspektive in der Anzeige 'Navigator' für die Serverkonfiguration auf **Server Configurations**, und wählen Sie anschließend **New** → **Server Instance and Configuration** aus.

- b. Ordnen Sie dem Server den Namen VideoCentralServer zu, und wählen Sie unter **Server instance type** die Option **WebSphere Servers** —> **WebSphere v4.0 Remote Server** aus. Jetzt wird die in Abbildung 3 dargestellte Anzeige aufgerufen.

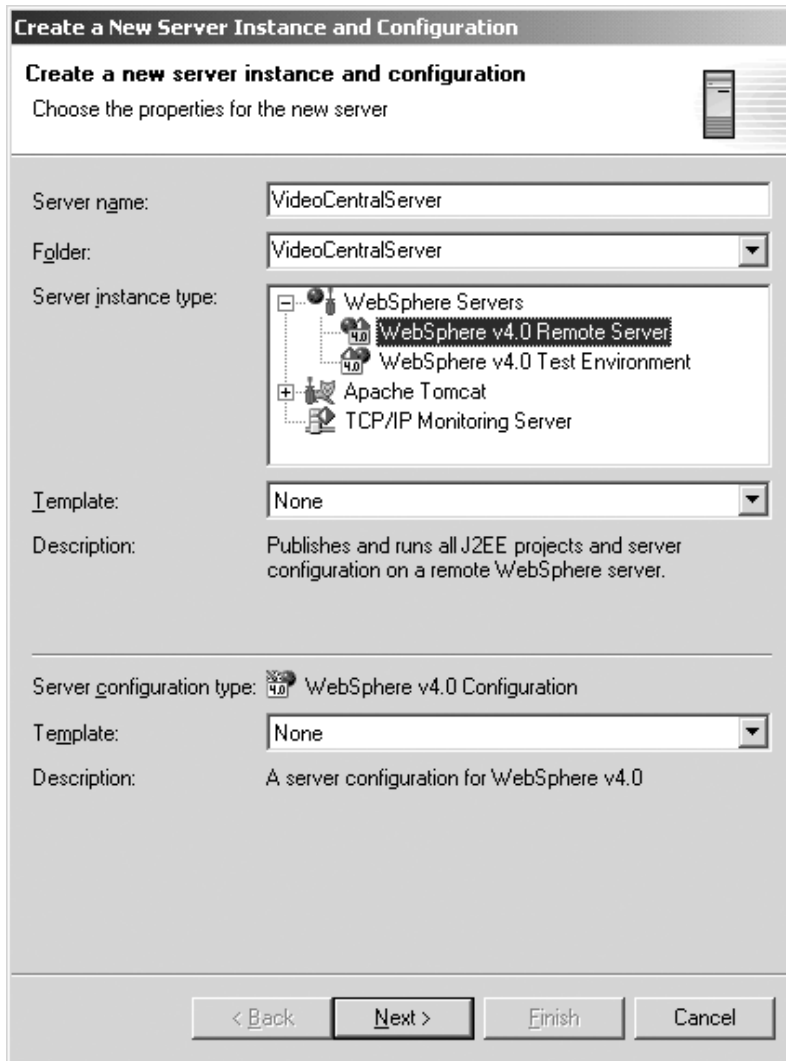


Abbildung 3. Erstellen und Konfigurieren eines neuen Serverexemplars

- c. Klicken Sie auf **Next**.
- d. Ersetzen Sie im Feld **Host Address** den Wert 127.0.0.1 durch localhost.

- e. Geben Sie im Feld **WebSphere installation directory** die Position ein, an der WebSphere Application Server 4.0 Advanced Edition Single Server installiert wurde (<WAS_install_dir>). Klicken Sie auf **Next**.
- f. Wählen Sie **Create a new remote file transfer instance** —> **Copy file transfer mechanism** aus. Klicken Sie auf **Next**.
- g. Geben Sie im Feld **Remote target directory** die Position ein, an der WebSphere Application Server 4.0 Advanced Edition Single Server installiert wurde (<WAS_install_dir>). Klicken Sie auf **Next**.
- h. Definieren Sie als HTTP-Portnummer 9080 an Stelle von 8080. Klicken Sie auf **Finish**. Jetzt wird in der Anzeige 'Navigator' die in Abbildung 4 dargestellte Anzeige aufgerufen.

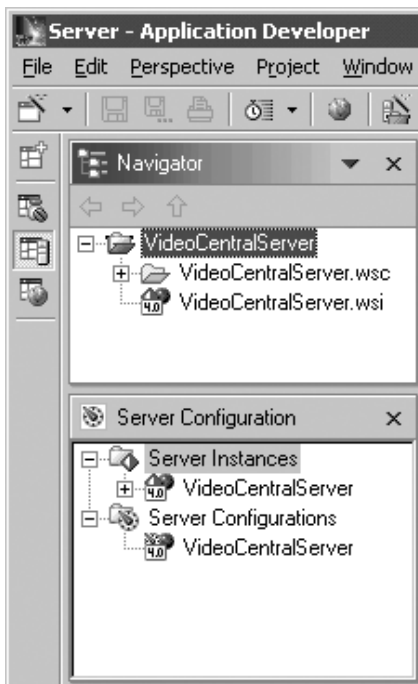


Abbildung 4. Erstelltes Serverexemplar

3. Klicken Sie in der Anzeige 'Server Configuration' im Ordner **Server Configurations** mit der rechten Maustaste auf **VideoCentralServer** —> **Open**.
 - a. Wählen Sie unter der Indexzunge 'General' die Option **Enable Administration Client** aus.
 - b. Nehmen Sie unter der Indexzunge 'EJB' die Auswahl der Option **Enable EJB test client** zurück.
 - c. Klicken Sie auf **File** —> **Save VideoCentralServer**.
 - d. Schließen Sie das Fenster **VideoCentralServer**.

4. Konfigurieren Sie die Datenquelle:
 - a. Wählen Sie im selben Fenster die Indexzunge 'Data source' aus, und heben Sie in der Liste der JDBC-Treiber den Eintrag **Db2JdbcDriver** hervor.
 - b. Fügen Sie die Beispieldatenbank hinzu:
 - 1) Klicken Sie auf den Knopf **Add** neben dem Feld **Data source defined in the JDBC driver selected above**.
 - 2) Geben Sie im Feld **Name** die Zeichenfolge `sample` ein.
 - 3) Geben Sie im Feld **JNDI name** die Zeichenfolge `jdbc/Sample` ein. (Hinweis: Im Feld für den JNDI-Namen muss die Groß-/Kleinschreibung beachtet werden. `Sample` muss im vorliegenden Fall deswegen mit großem Anfangsbuchstaben geschrieben werden.)
 - 4) Geben Sie im Feld **Database name** die Zeichenfolge `sample` ein.
 - 5) Geben Sie im Feld **Default User ID** die Zeichenfolge `ivc` ein.
 - 6) Geben Sie im Feld **Default User password** die Zeichenfolge `VideoCentral` ein.
 - 7) Klicken Sie auf **OK**.

Anmerkung: Die Video Central-Anwendung stellt Verbindungen zu DB2 her und verwendet hierzu die Benutzer-ID und das Kennwort, das in `VideoCentral.properties` gespeichert ist.

Jetzt wird die in Abbildung 5 dargestellte Anzeige aufgerufen.

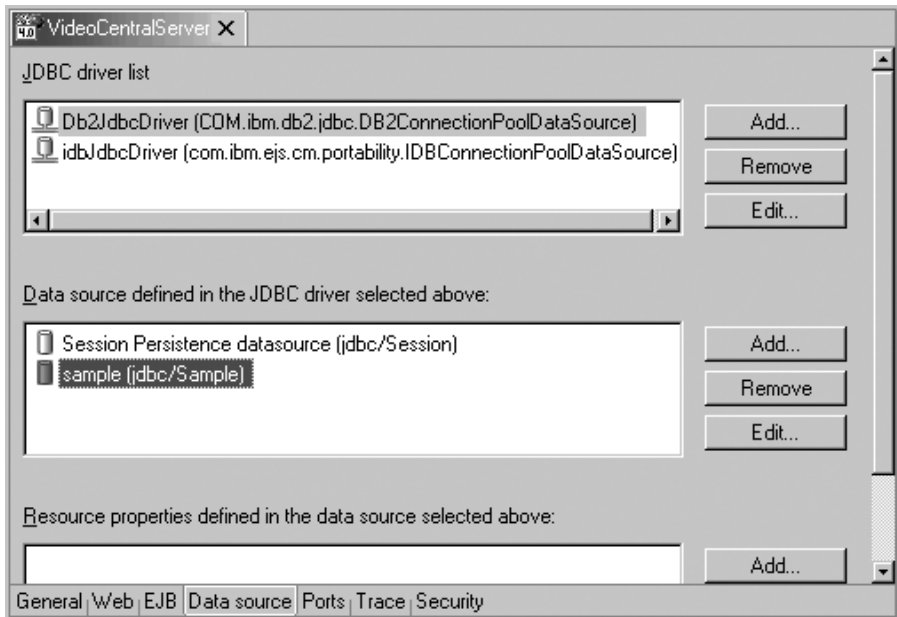


Abbildung 5. Konfigurieren der Datenquelle

- c. Schließen Sie das Fenster VideoCentralServer, indem Sie auf das X klicken und anschließend **Yes** auswählen, wenn Sie in einem daraufhin eingblendeten Fenster gefragt werden, ob die Änderungen gespeichert werden sollen.
5. Klicken Sie in der Anzeige 'Server Configuration' im Ordner **Server Instances** mit der rechten Maustaste auf **VideoCentralServer** → **Open**.
 - a. Wählen Sie die Indexzunge 'Paths' aus, und klicken Sie auf **Add External JARs**. Fügen Sie folgende Komponenten hinzu:
 - <DB2_install_dir>\java\db2java.zip
 - <DB2_install_dir>\tools\databean.jar
 - <WAS_install_dir>\lib\j2ee.jar

Anmerkung: Die Datei db2java.zip umfasst eine Gruppe von Klassen, die JDBC 2.1-Unterstützung für den Zugriff auf DB2-Datenbanken bereitstellen. Diese Klassen kommunizieren mit dem DB2-Server über eine JDBC-Treiberarchitektur des Typs 2.

- b. Schließen Sie das Fenster VideoCentralServer, indem Sie auf das X klicken und anschließend **Yes** auswählen, wenn Sie in einem daraufhin eingblendeten Fenster gefragt werden, ob die Änderungen gespeichert werden sollen.

Task 2: Definieren des Webprojekts VideoCentral und Importieren des Quellcodes

Erstellen des Webprojekts VideoCentral

1. Erstellen Sie das Webprojekt VideoCentral:
 - a. Klicken Sie in der Webperspektive auf **File** → **New** → **Web project**, um ein neues Webprojekt zu erstellen. Ordnen Sie dem Projekt den Namen VideoCentral und dem Unternehmensanwendungsprojekt den Namen VideoCentralEAR zu. Klicken Sie auf **Next**.
 - b. Daraufhin wird die Anzeige **Module Dependencies** aufgerufen. Übernehmen Sie alle Standardeinstellungen. Klicken Sie auf **Next**.
 - c. Daraufhin wird die Anzeige **Define Java Build Settings** aufgerufen. Wählen Sie die Indexzunge 'Libraries' aus, und klicken Sie auf **Add External JARs**. Fügen Sie die folgenden externen JAR-Komponenten hinzu, die im Dialogfenster **JAR Selection** angezeigt werden:
 - `<WSAD_install_dir>\plugins\com.ibm.etools.webservices\runtime\soap.jar`
 - `<DB2_install_dir>\tools\databean.jar`

Anmerkung: Die Datei `soap.jar` ist für Video Central erforderlich, um die von den Web-Services angeforderte Syntaxanalyse der SOAP-Nachrichten durchzuführen.

Die Datei `databean.jar` umfasst eine Gruppe von Klassen, die von IBM zur Vereinfachung des DB2-Zugriffs über Java bereitgestellt werden.

Jetzt wird die in Abbildung 6 dargestellte Anzeige aufgerufen.

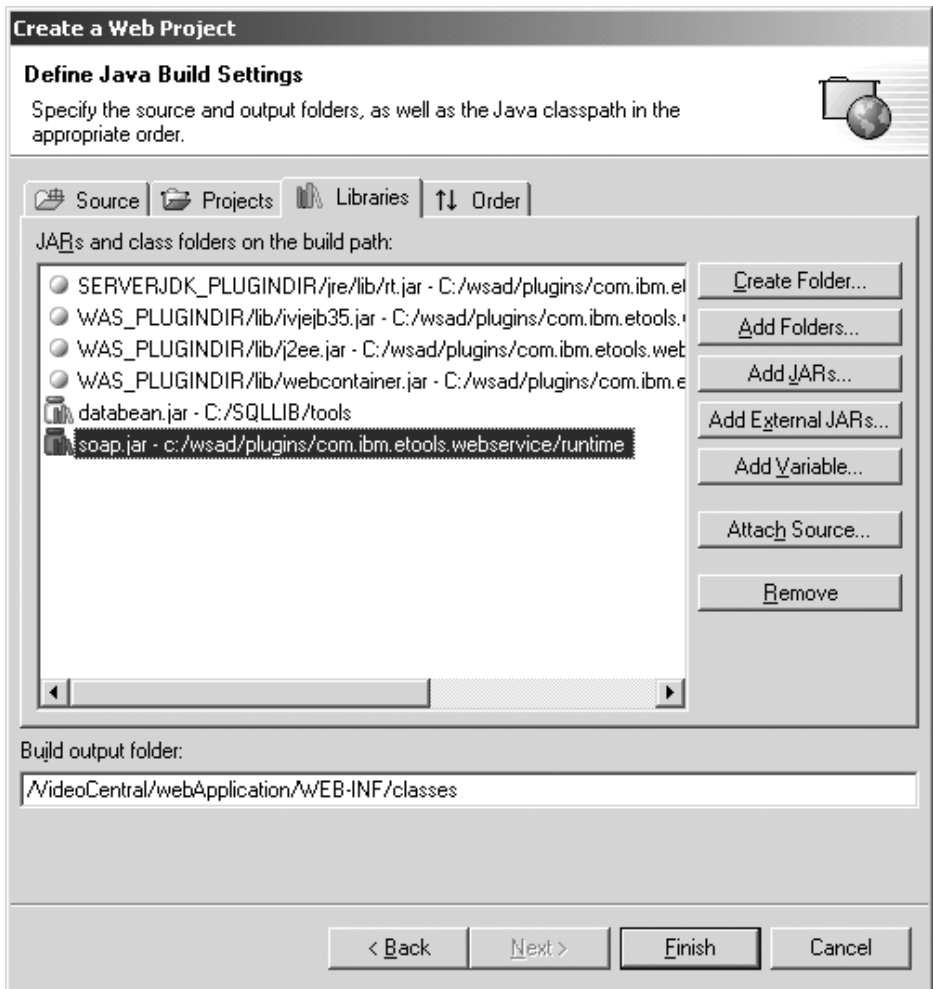


Abbildung 6. Hinzufügen externer JAR-Dateien

d. Klicken Sie auf **Finish**.

2. Fügen Sie das Webprojekt VideoCentral zum Server hinzu:
 - a. Erweitern Sie in der Anzeige 'Server Configuration' die Anzeige für **Server Configurations**, und klicken Sie mit der rechten Maustaste auf **VideoCentralServer**. Wählen Sie anschließend **Add Project** → **VideoCentralEAR** aus.

Jetzt wird in der Anzeige 'Server Configuration' die in Abbildung 7 dargestellte Anzeige aufgerufen.

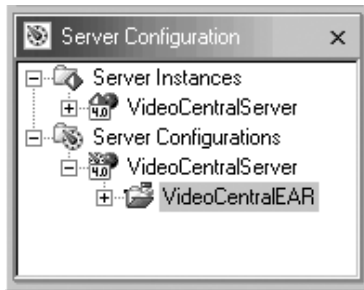


Abbildung 7. Hinzufügen des Webprojekts zum Server

- b. Klicken Sie in der Anzeige 'Navigator' mit der rechten Maustaste auf das Webprojekt VideoCentral, und wählen Sie anschließend **Properties** → **Server Preferences** aus. Heben Sie den Eintrag **VideoCentralServer** im Fenster hervor, klicken Sie auf **Apply** und dann auf **OK** (siehe Abbildung 8).

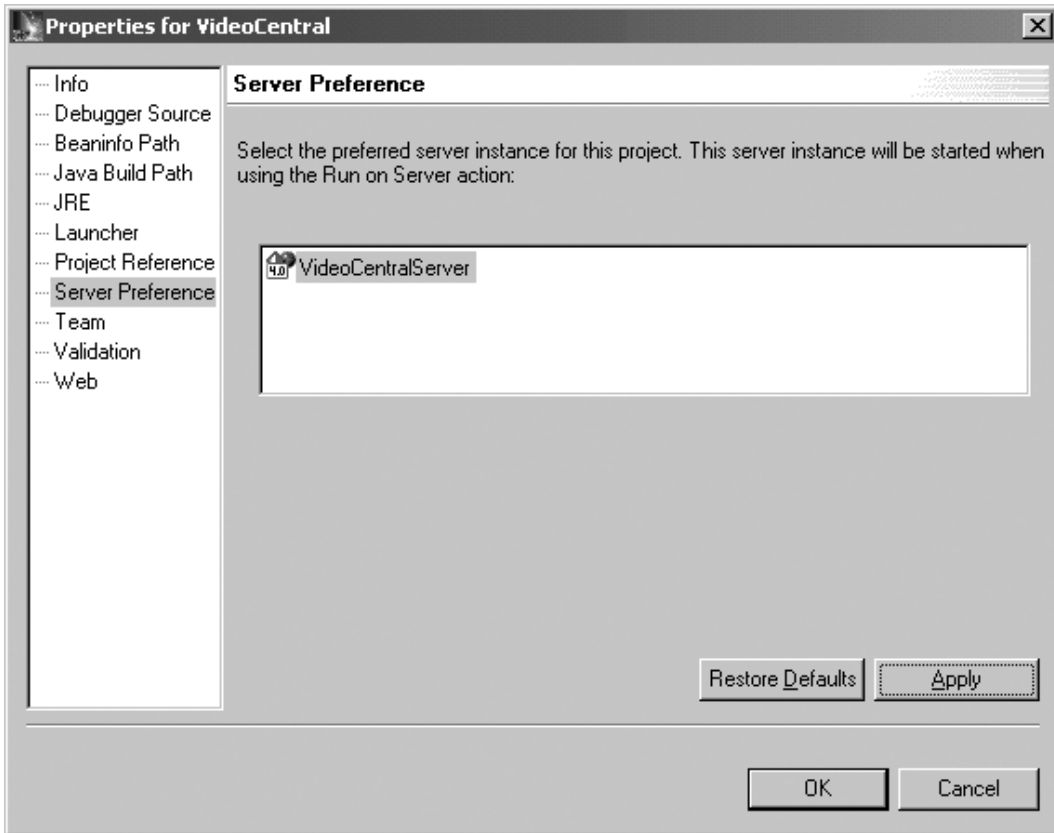


Abbildung 8. Hinzufügen des Webprojekts zum Server (Fortsetzung)

Importieren von VideoCentral.jar in WebSphere Studio Application Developer

1. Erweitern Sie die Sicht des neu erstellten Webprojekts **VideoCentral**, und heben Sie **source** (vgl. Abbildung 9) hervor.

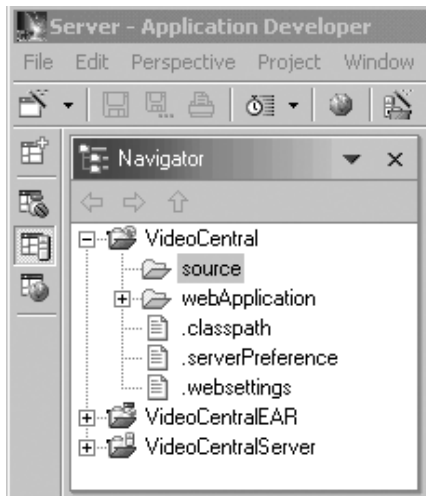


Abbildung 9. Vor dem Import der JAR-Dateien

2. Wählen Sie **File** → **Import** aus.
3. Wählen Sie als Importquelle **Zip file** aus. Klicken Sie auf **Next**.
4. Suchen Sie nach
<DB2_install_dir>\tutorials\VideoCentral\lib\VideoCentral.jar.
5. Klicken Sie auf den Knopf **Select All** (siehe Abbildung 10).

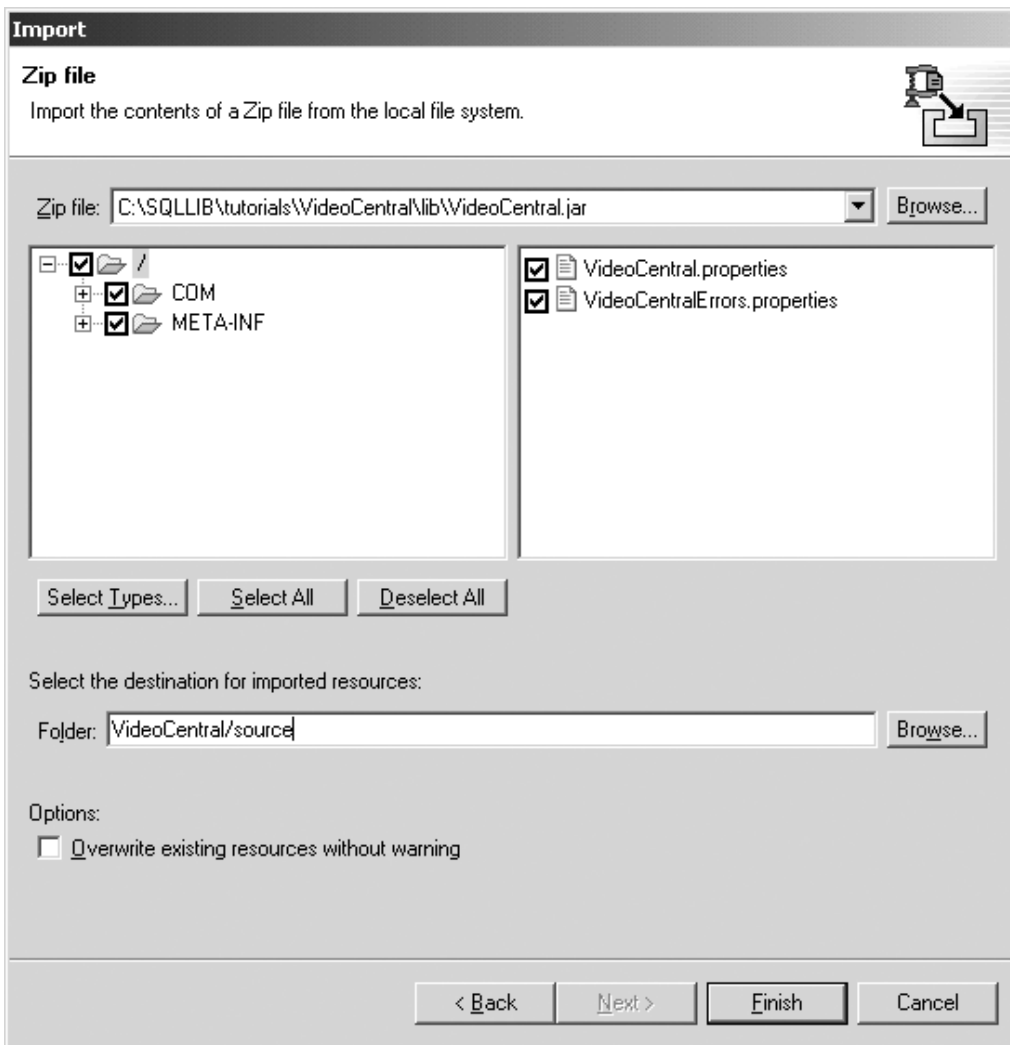


Abbildung 10. Auswahl der zu importierenden Ressourcen

6. Klicken Sie auf **Finish**.

Die Dateien werden nun in WebSphere Studio Application Developer importiert und Sie können in der Aktionsleiste sehen, dass die Dateien kompiliert werden.

7. Nach dem Import der Dateien in das Webprojekt VideoCentral müssen Sie die Sicht des Quellenordners erweitern. Dieser muss die in Abbildung 11 dargestellte Hierarchiestruktur aufweisen.

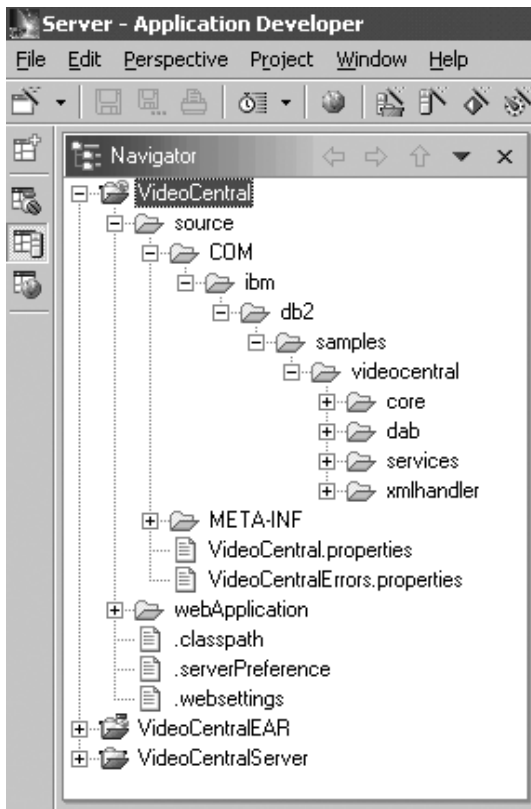


Abbildung 11. Nach der Vorbereitung des Video Central-Projekts

Zusammenfassung der Lerneinheit

In diesem Kapitel wurden die folgenden Themen erläutert:

- WebSphere Studio Application Developer
- Erstellen einer WAS-Serverreferenz in WebSphere Studio Application Developer zum Implementieren und Testen von Anwendungen
- Importieren des Video Central-Codes in WebSphere Studio Application Developer

Nächster Schritt

Als Nächstes wird das Design der Video Central-Anwendung detailliert beschrieben.

Kapitel 5. Prüfen des Video Central-Designs

Dauer

Zum Durcharbeiten der vorliegenden Lerneinheit werden ca. 20 Minuten benötigt.

Übersicht

Im vorliegenden Kapitel wird das Design und die Implementierung von Video Central in Form der folgenden Softwarekomponentenschichten erläutert:

1. Webschnittstellenschicht (WIL = Web Interface Layer)
 2. Unternehmenslogikschicht (BLL = Business Logic Layer)
 3. Datenzugriffsschicht (Data Access Layer = DAL)
-

Video Central-Design

Im vorliegenden Abschnitt wird das Design von Video Central erläutert. Zum einfacheren Verständnis werden im Folgenden verschiedene Klassen- und Sequenzdiagramme dargestellt.

Webschnittstellenschicht (WIL)

Die Webschnittstellenschicht (WIL = Web Interface Layer) stellt die verschiedenen Web-Services, die von Video Central unterstützt werden, für die Konsumenten zur Verfügung. Web-Services können über eine höhere Abstraktionsebene bereitgestellt werden. Das Design von Video Central selbst umfasst diese Abstraktionsebene für die zugehörigen Web-Services nicht, verwendet auf dieser Ebene jedoch die SOAP-Laufzeitkomponente.

Entwickeln der Unternehmenslogikschicht (BLL)

Die in Video Central implementierten Klassen- und Sequenzdiagramme der Java-Klassen werden hier bereitgestellt. Die eigentlichen Java-Dateien befinden sich in den Dateien

<DB2_install_dir>\tutorials\VideoCentral\lib\VideoCentral.jar und
<DB2_install_dir>\tutorials\VideoCentral\lib\VideoClient.jar, die im vorliegenden Lernprogramm zu Informationszwecken aufgeführt sind.

Prüfen der Video Central-Implementierung

Die folgenden Abschnitte enthalten umfassendere Informationen zur Unternehmenslogikschicht (BLL = Business Logic Layer) von Video Central.

Klassendiagramm für Video Central: Abbildung 12 zeigt das Übersichts- klassendiagramm für Video Central.

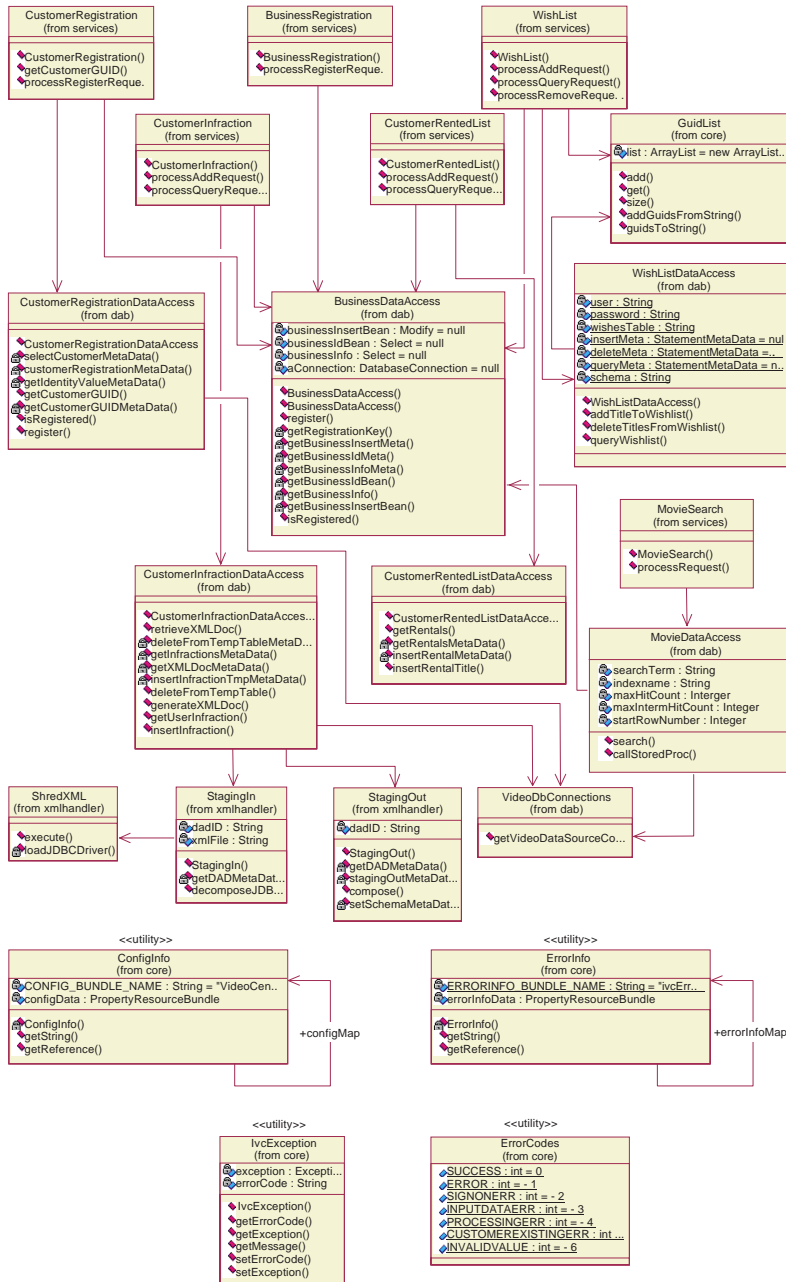


Abbildung 12. Video Central - Übersichtsclassendiagramm

Klassendiagramm zum Web-Service für Kundenverstöße: Jeder Web-Service wird im Prinzip auf die selbe Weise implementiert. Der Web-Service für Kundenverstöße nimmt hierbei allerdings eine Ausnahmestellung ein, da er den Einsatz von DB2 XML Extender demonstriert. Auf Grund dieser Tatsache wird er hier detaillierter behandelt, um zu zeigen, wie auf gespeicherte XML-Prozeduren zugegriffen werden kann und wie diese eingesetzt werden können. Das folgende Klassendiagramm erläutert die Implementierung des Web-Services für Kundenverstöße (siehe Abbildung 13).

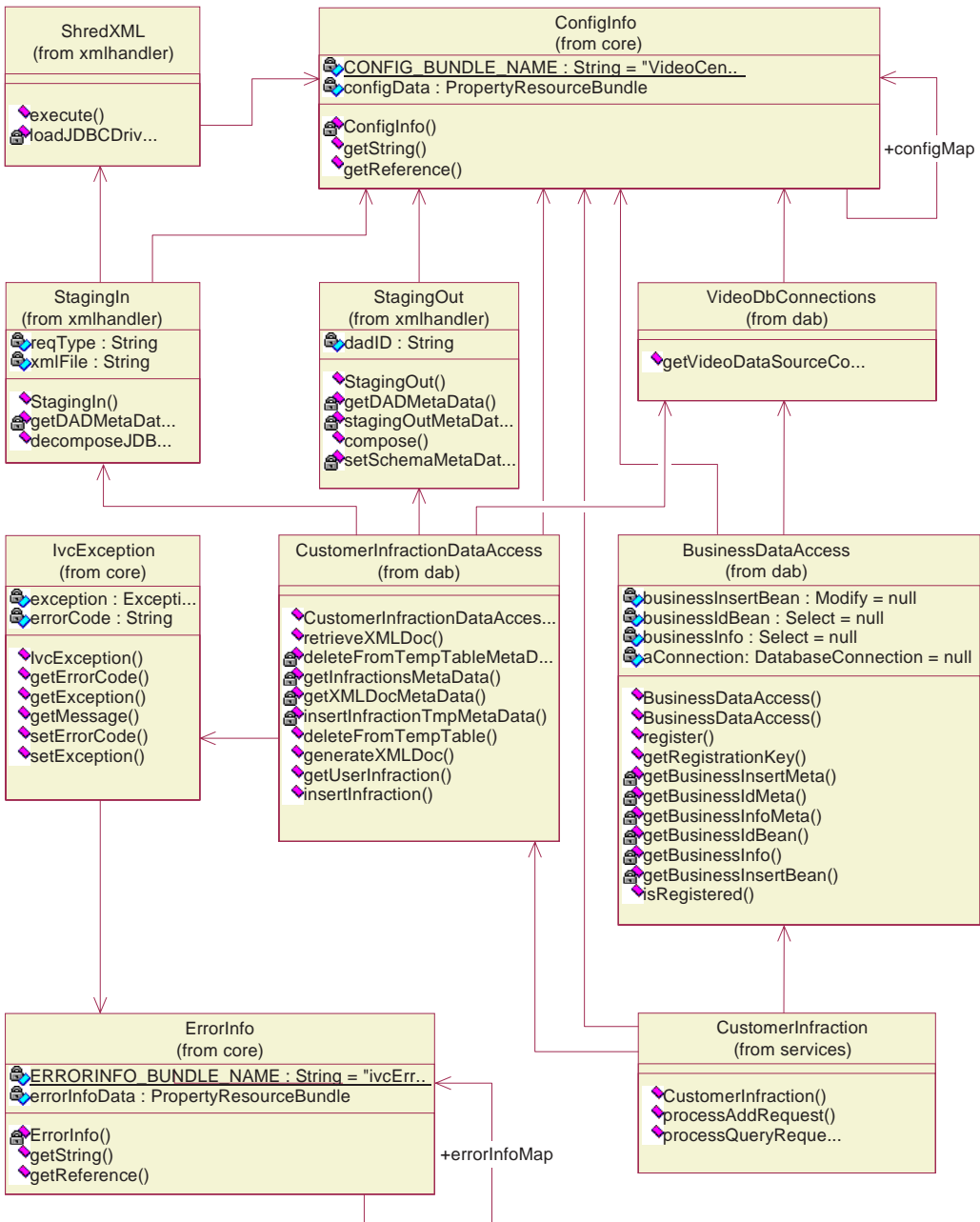


Abbildung 13. Klassendiagramm zum Web-Service für Kundenverstöße

In Abbildung 13 steht CustomerInfracation für die Unternehmenslogikschicht des Web-Services für Kundenverstöße. Während CustomerInfracationDataAccess, StagingIn, ShredXML und StagingOut zusammen die Datenzugriffsschicht bilden, die auch DB2 XML Extender für die Zusammensetzung und Zerlegung von XML-Daten umfasst, stellt VideoDbConnections die Funktionalität zum Abrufen von Datenbankverbindungen von einer Datenquelle bereit.

BusinessDataAccess ist in jedem Web-Service für die Unternehmensprüfung eingebunden.

Klassendiagramm zum Web-Service für die Wunschliste: Die Implementierung des Web-Services für die Wunschliste verwendet Data Access-JavaBeans (im Paket `com.ibm.db.*`). Alle anderen Services wurden mit IBM Data Access-Beans (im Paket `com.ibm.db.uibeans.*`) implementiert. Die mit DB2 UDE bereitgestellte Datei `databean.jar` enthält die Klassen, die für die Datenzugriffsschicht von Video Central erforderlich sind. Das folgende Klassendiagramm erläutert die Implementierung des Web-Services für Wunschlisten (siehe Abbildung 14).

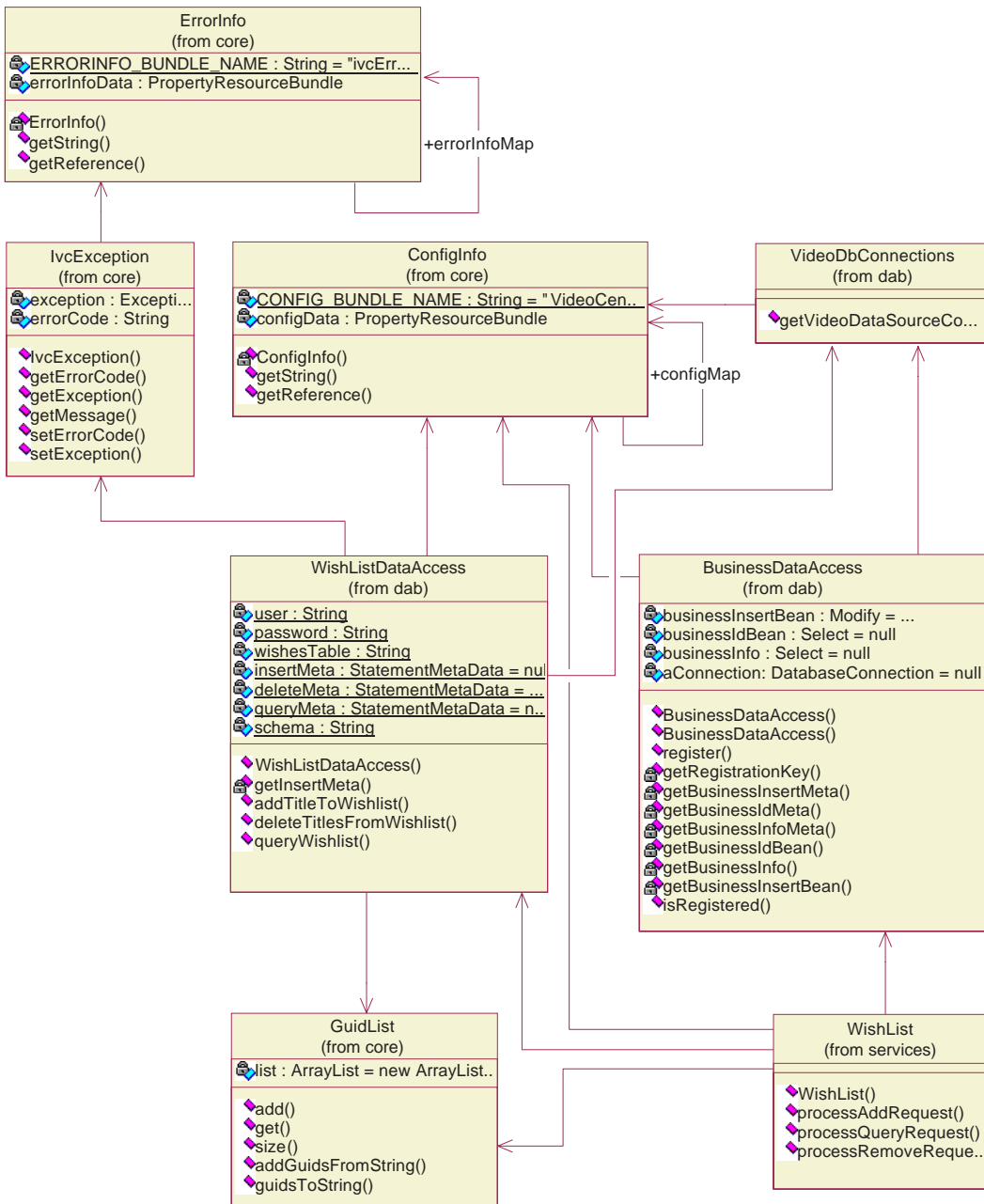


Abbildung 14. Klassendiagramm zum Web-Service für Wunschlisten

Sequenzdiagramm zum Hinzufügen von Kundenverstößen: Da alle Web-Services im Prinzip auf die gleiche Weise implementiert werden, wurden im vorliegenden Lernprogramm nur die aussagekräftigsten der Sequenzdiagramme für die sechs Services dargestellt. Das Sequenzdiagramm für AddCustomerInfraction ist von besonderem Interesse, weil es die gespeicherte Prozedur DXXSHREXML für die Zerlegung von DB2 XML Extender verwendet (siehe Abbildung 15). Wenn Sie die Methode execute() in der Klasse ShredXML betrachten, können Sie die Verwendung dieser gespeicherten Prozedur feststellen.

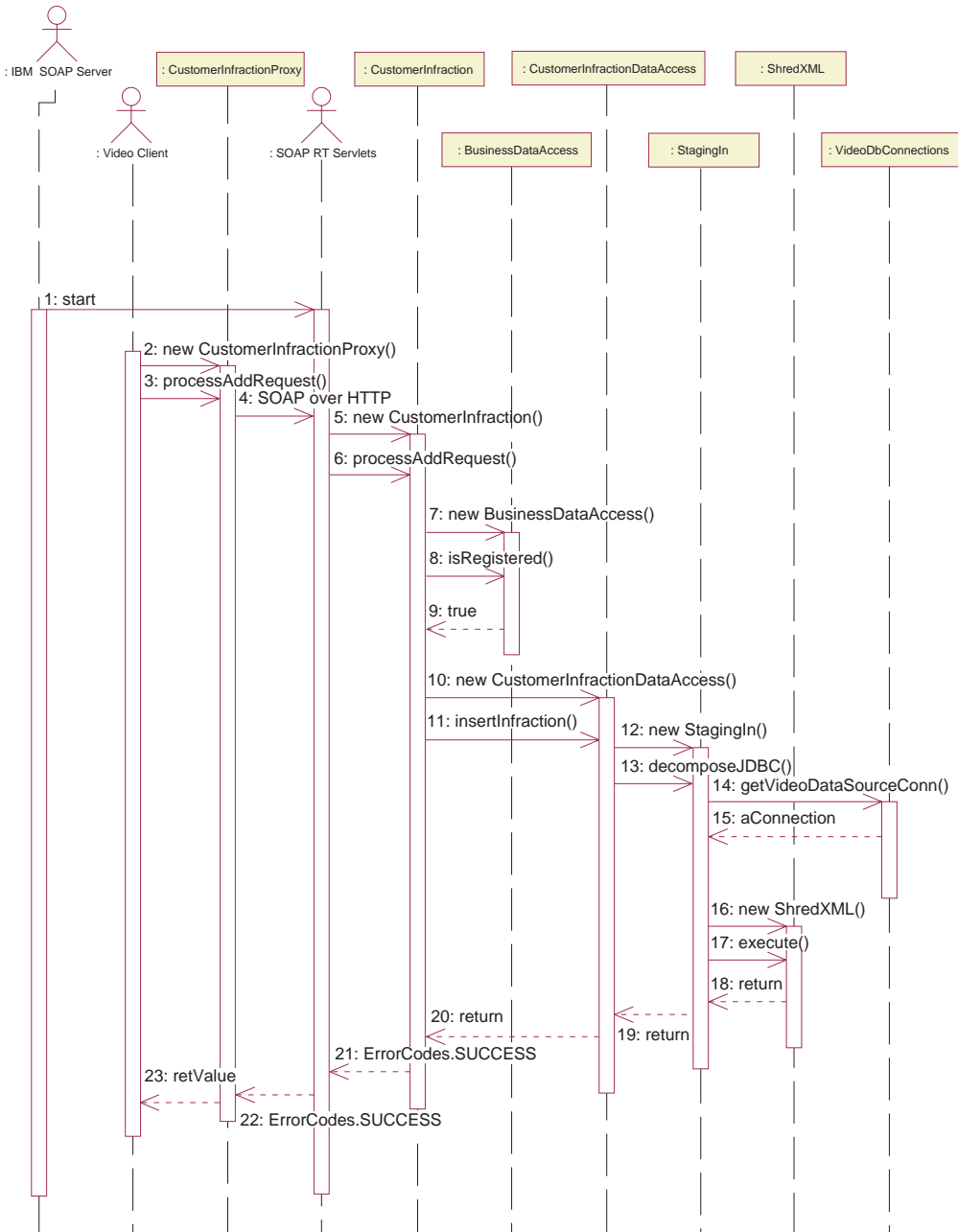


Abbildung 15. Sequenzdiagramm zum Hinzufügen von Kundenverstößen

Sequenzdiagramm zur Abfrage von Kundenverstößen: Das Sequenzdiagramm zur Abfrage von Kundenverstößen (QueryCustomerInfraction) ist ebenfalls von besonderem Interesse, da es die gespeicherte Prozedur DXXGENXML zur Zusammensetzung von DB2 XML Extender (siehe Abbildung 16) verwendet. Wenn Sie die Methode compose() in der Klasse StagingOut betrachten, können Sie die Verwendung dieser gespeicherten XML Extender-Prozedur für die Zusammensetzung feststellen.

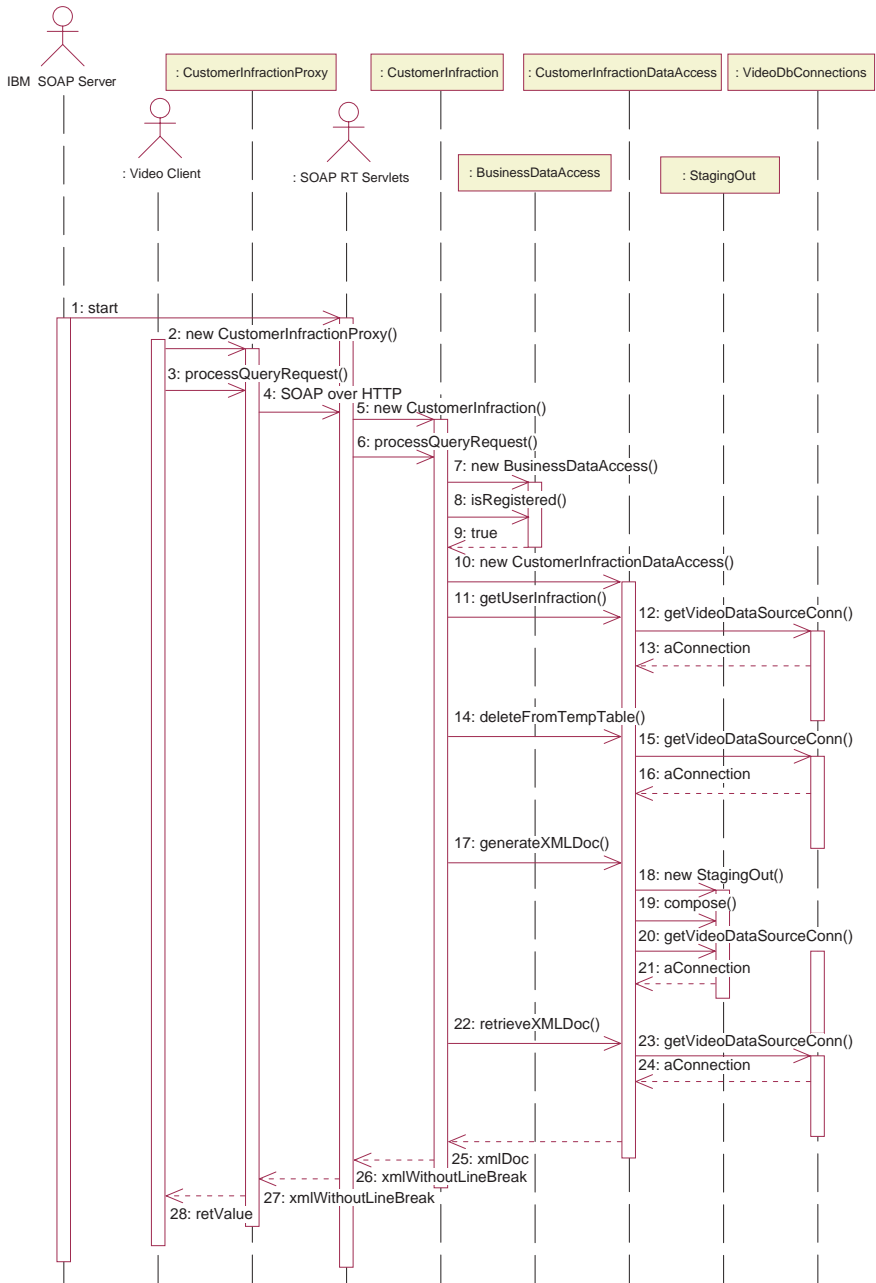


Abbildung 16. Sequenzdiagramm zur Abfrage von Kundenverstößen

Sequenzdiagramm zum Hinzufügen einer Wunschliste: Das Sequenzdiagramm zum Hinzufügen einer Wunschliste wird hier dargestellt, um den typischen Ablauf eines Web-Services zu erläutern. Der Service für die Wunschliste wurde mit einer niedrigeren Ebene von Data Access-Klassen als den Data Access-Beans im Paket com.ibm.db.* implementiert, der Sequenzablauf des Services ist jedoch ähnlich. Weitere Informationen hierzu enthält Abbildung 17. Die Schritte 10 bis 14 dienen zum Einfügen eines Titels für eine Kundenwunschliste über die Datenzugriffsschicht.

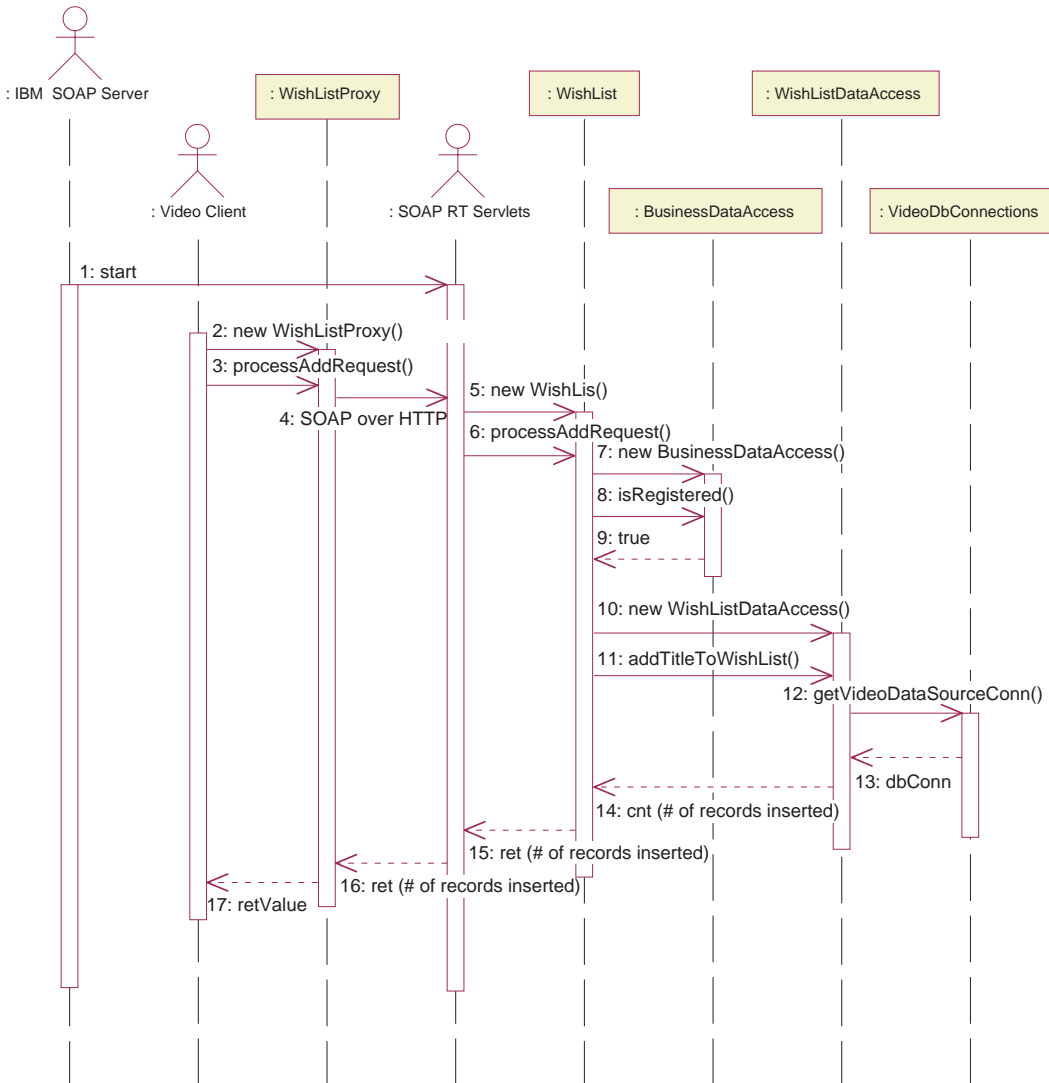


Abbildung 17. Sequenzdiagramm zum Hinzufügen einer Wunschliste

Erstellen von Datenmodellen

Das Datenmodell der für Video Central erforderlichen Tabellen wird im Folgenden beschrieben. Weitere Einzelheiten hierzu finden Sie in den Scriptdateien im Verzeichnis <DB2_install_dir>\tutorials\VideoCentral\db.

Übersicht zum Datenmodell

Abbildung 18 und Abbildung 19 zeigen die von Video Central verwendeten Tabellen.

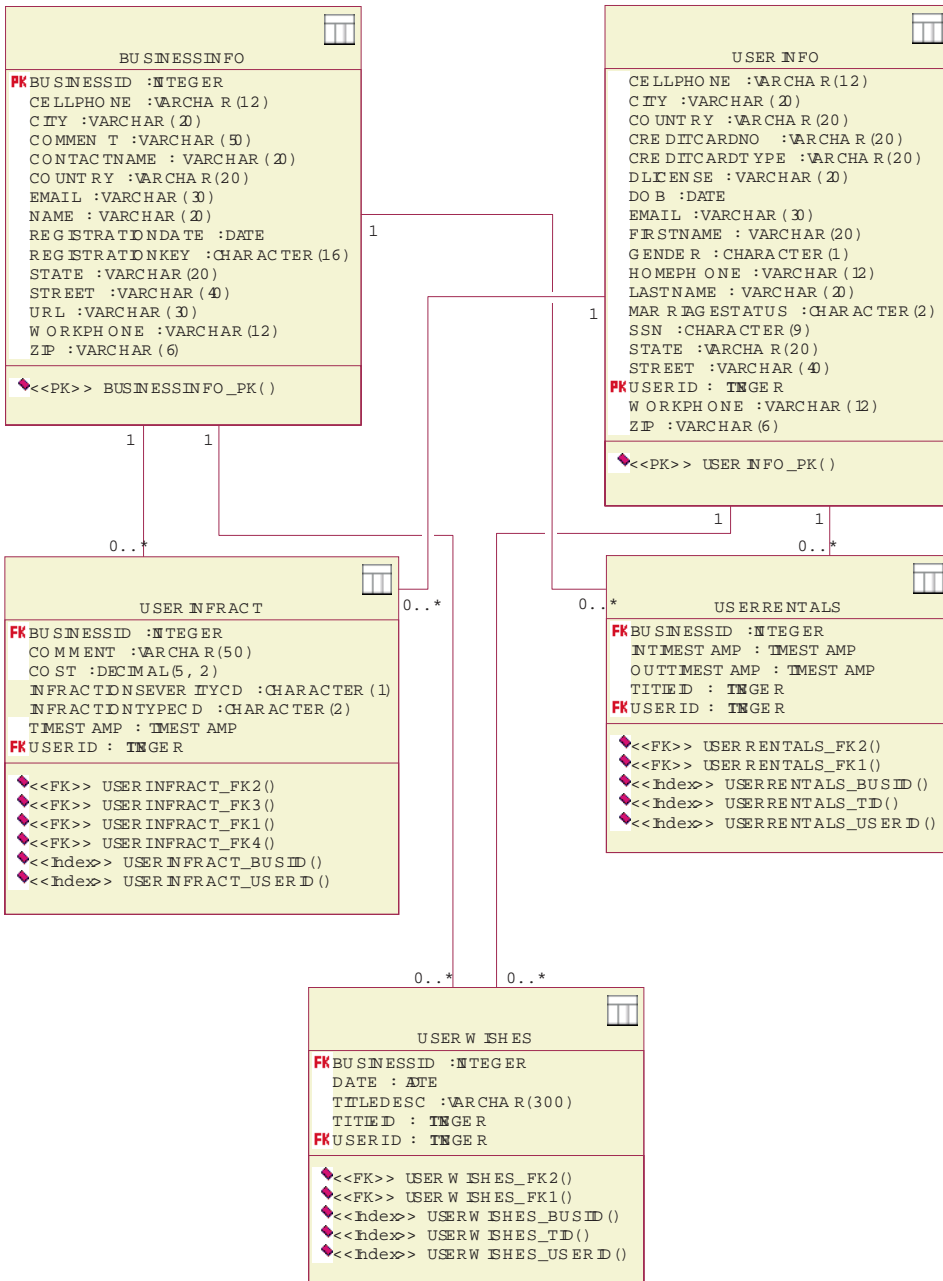


Abbildung 18. Diagramm des Datenmodells - Teil 1

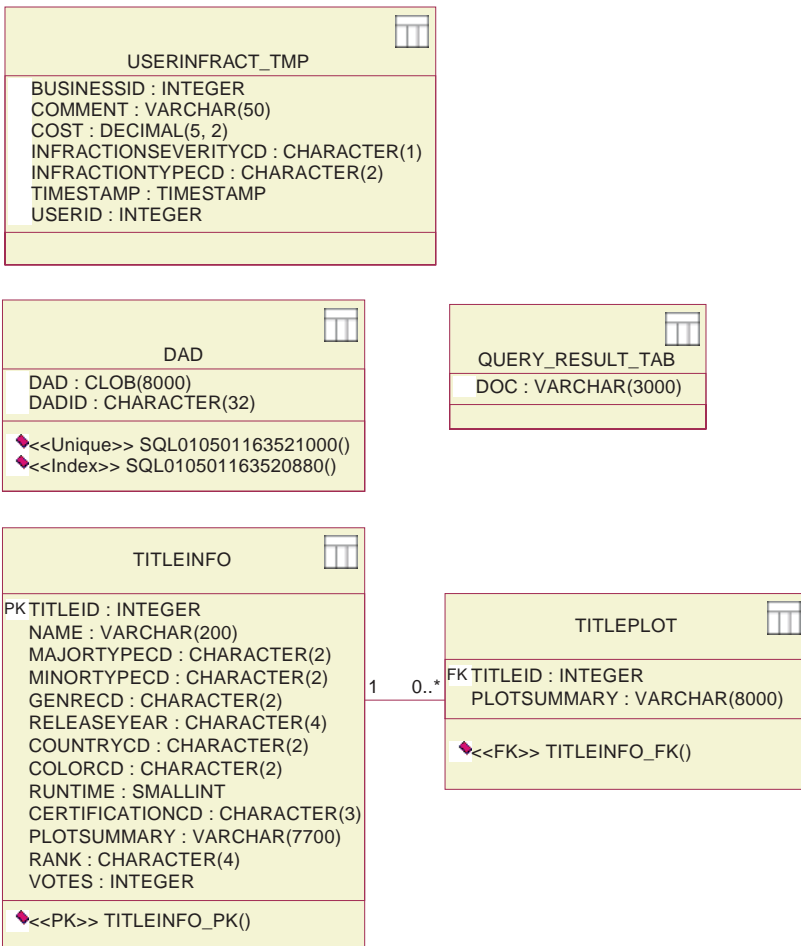


Abbildung 19. Diagramm des Datenmodells - Teil 2

In der Tabelle DAD werden die DAD-Dateien (DAD = Document Access Definition) gespeichert, die zur Zusammensetzung und Zerlegung mit DB2 XML Extender verwendet werden. Die Tabellen USERINFRACT_TMP und QUERY_RESULT_TAB dienen zum Speichern der Berichte zu den Kundenverstößen, die für die Abfragen dieser Verstöße generiert werden. Diese beiden Tabellen können entfernt werden, wenn die Funktion DxxGenXMLC1ob von DB2 XML Extender verwendet wird. Die Tabellen TITLEINFO und TITLEPLOT werden zum Speichern von Informationen zum Filmtitel und zur Handlung benutzt. Die Indizes der Spalten TITLEINFO.NAME und TITLEPLOT.PLOTSUMMARY werden von DB2 Net Search Extender zum Bereitstellen des Web-Services für die Filmtitelsuche benutzt.

Zusammenfassung der Lerneinheit

In diesem Kapitel wurden grundlegende Kenntnisse zum Design und zum Datenmodell von Video Central erläutert.

Nächster Schritt

Als Nächstes werden die Web-Services von WebSphere Studio Application Developer im WebSphere Application Server implementiert.

Kapitel 6. Implementieren von Video Central

Dauer

Zum Durcharbeiten der vorliegenden Lerneinheit werden ca. 20 Minuten benötigt.

Voraussetzungen

- WebSphere Application Server 4.0, Advanced Edition Single Server
 - IBM HTTP-Server, Version 1.3.12. Dieses Produkt kann unter der Adresse <http://www.ibm.com/software/webservers/httpservers/download.html> abgerufen werden.
 - WebSphere Studio Application Developer und die Datei VideoCentral.jar (das Importieren von VideoCentral.jar in WebSphere Studio Application Developer wird in Kapitel 4 beschrieben).
-

Übersicht

Im vorliegenden Kapitel werden Sie den Code von Video Central im WebSphere Application Server (zuvor in Kapitel 4 definiert) in der Umgebung von WebSphere Studio Application Developer implementieren.

Video Central-Tasks

Sie werden nun lernen, wie ein Web-Service von WebSphere Studio Application Developer im WebSphere Application Server implementiert wird.

Task 1: Vorbereiten der Umgebung für das Implementieren von Video Central vom WebSphere Studio Application Developer

1. Prüfen Sie, ob DB2 und der HTTP-Server gestartet sind:
 - a. Rufen Sie die Windows-Services auf, und bestätigen Sie, dass DB2 und IBM HTTP Server gestartet wurden. Andernfalls müssen Sie diese Produkte nun starten.
 - b. Öffnen Sie einen Webbrowser und geben Sie als URL localhost an. Jetzt wird die in Abbildung 20 dargestellte Anzeige aufgerufen.



Welcome to the IBM HTTP Server

powered by Apache

*Everything you need to start
and use this server begins here...*

- **Configure server**
- **View documentation**
- **Visit our Web site**



Abbildung 20. Prüfen des HTTP-Servers

2. Wählen Sie die Anzeige 'Servers' in der Serverperspektive von WebSphere Studio Application Developer aus. Klicken Sie in dieser Anzeige mit der rechten Maustaste auf VideoCentralServer, und wählen Sie anschließend **Start** aus. Wählen Sie die Anzeige 'Console' aus, und prüfen Sie, ob die Nachricht "Server Default Server open for e-business" (vgl. Abbildung 21) vorhanden ist.


```
Console
A DYNA0011E: Servlet cache file dynacache.xml not found; c
A SRVE0169I: Loading Web Module: admin.
I SRVE0091I: [Servlet LOG]: JSP 1.1 Processor: init
I SRVE0091I: [Servlet LOG]: SimpleFileServlet: init
I SRVE0091I: [Servlet LOG]: action: init
A SRVE0169I: Loading Web Module: IBMVideoCentral.
I SRVE0091I: [Servlet LOG]: JSP 1.1 Processor: init
I SRVE0091I: [Servlet LOG]: SimpleFileServlet: init
I SRVE0091I: [Servlet LOG]: InvokerServlet: init
A SRVE0169I: Loading Web Module: IBMVideoCentralClient.
I SRVE0091I: [Servlet LOG]: JSP 1.1 Processor: init
I SRVE0091I: [Servlet LOG]: SimpleFileServlet: init
I SRVE0091I: [Servlet LOG]: InvokerServlet: init
A SRVE0171I: Transport http is listening on port 9,080.
A SRVE0171I: Transport http is listening on port 9,090.
U *** Server has started ***
A WSWR0023I: Server Default Server open for e-business
```

Abbildung 21. Ausführung des WAS Admin Server prüfen

3. Wählen Sie die Anzeige 'Servers' in der Serverperspektive von WebSphere Studio Application Developer aus. Klicken Sie in dieser Anzeige mit der rechten Maustaste auf VideoCentralServer, und wählen Sie anschließend **Stop** aus.

Anmerkung: VideoCentralServer wird gestoppt, weil der Server während der Implementierung von Web-Services automatisch gestartet wird. Wurde der Server bereits gestartet, wird die Fehlermeldung port in use ausgegeben.

Task 2: Implementieren der Web-Services im WebSphere Application Server

Erstellen und Implementieren der (Java-Bean-basierten) Web-Services mit WebSphere Studio Application Developer

1. Implementieren Sie die Web-Services von WebSphere Studio Application Developer in WebSphere Application Server:
 - a. Auswahl des Web-Service-Typs
Wechseln Sie in WebSphere Studio Application Developer in die Webperspektive, erweitern Sie die Sicht des Webprojekts VideoCentral, und wählen Sie
VideoCentral/source/COM/ibm/db2/samples/videocentral/services/
BusinessRegistration.java
als ersten Web-Service aus, der implementiert werden soll (vgl. Abbildung 22).

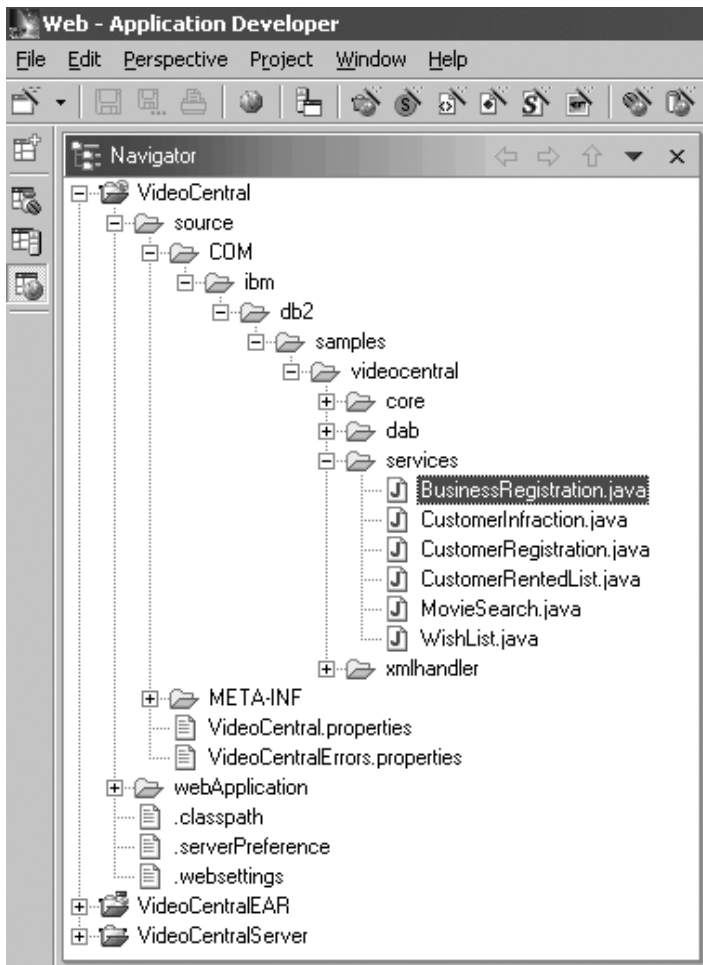


Abbildung 22. Auswahl des zu implementierenden Web-Services

- b. Klicken Sie mit der rechten Maustaste auf **BusinessRegistration.java** —> **New** —> **Other**, und in dem daraufhin angezeigten Fenster **New** auf **Web service**. Übernehmen Sie alle Standardeinstellungen, und klicken Sie auf **Next**.

- Bestätigen Sie im Fenster **Web Services Java Bean Selection**, dass die Bean mit der in **Schritt 1** (vgl. Abbildung 23) getroffenen Auswahl übereinstimmt.

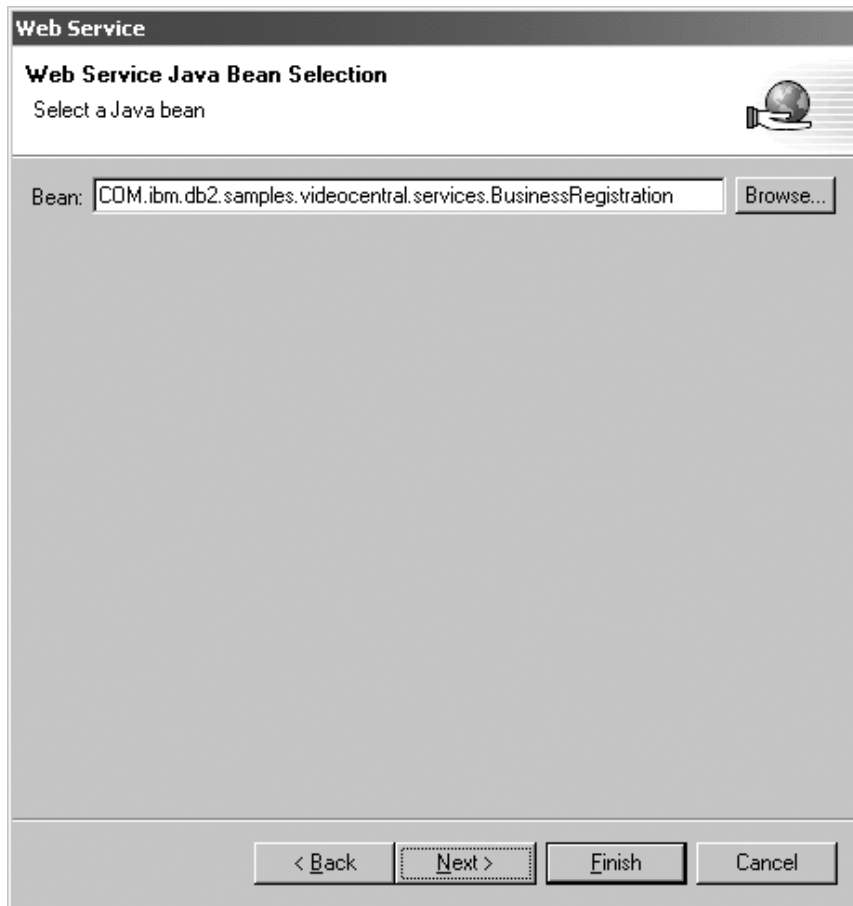
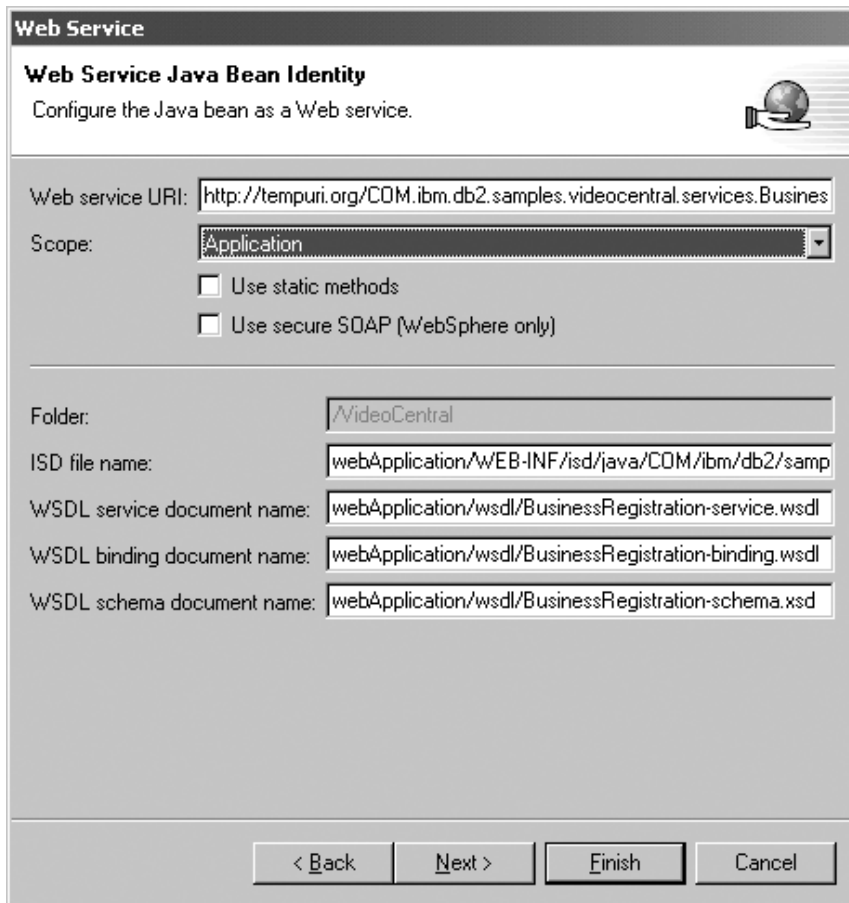


Abbildung 23. Zu implementierende Java-Bean

Klicken Sie auf **Next**.

3. Konfigurieren Sie die Java-Bean als Web-Service. Übernehmen Sie die Standardeinstellungen, und klicken Sie auf **Next** (vgl. Abbildung 24).



Web Service

Web Service Java Bean Identity

Configure the Java bean as a Web service.

Web service URI:

Scope:

Use static methods

Use secure SOAP (WebSphere only)

Folder:

ISD file name:

WSDL service document name:

WSDL binding document name:

WSDL schema document name:

< Back Next > Finish Cancel

Abbildung 24. Konfigurieren der Java-Bean als Web-Service

4. Übernehmen Sie die Auswahl aller Methoden (vgl. Abbildung 25).

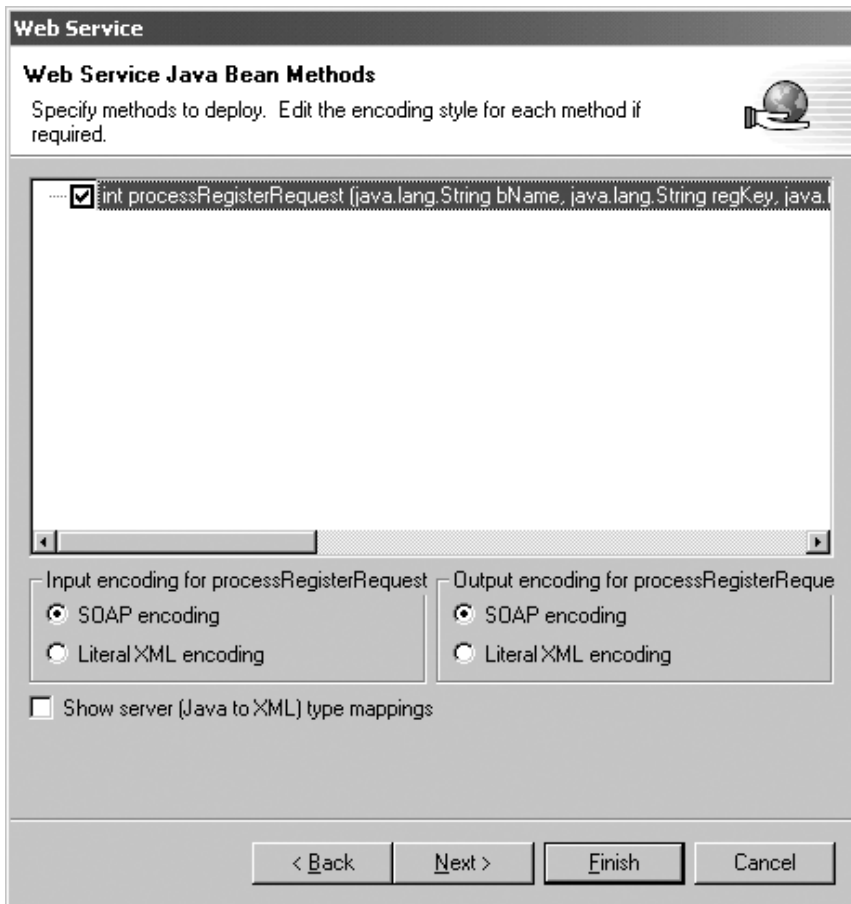


Abbildung 25. Auswahl der zu implementierenden öffentlichen Methoden

Klicken Sie auf **Next**.

- Ersetzen Sie auf der Seite **Web Service Binding Proxy Generation** im Feld **Class** den Wert:

```
proxy.soap.COM.ibm.db2.sample.videocentral.services.
```

```
BusinessRegistrationProxy
```

durch

```
COM.ibm.db2.samples.videoclient.proxies.BusinessRegistrationProxy
```

Weitere Informationen hierzu enthält Abbildung 26. Auf diese Weise kann die Proxyklasse im Paket `COM.ibm.db2.samples.videoclient.proxies.*` im Projekt VideoCentral erstellt werden.



Abbildung 26. Web Service Binding Proxy Generation

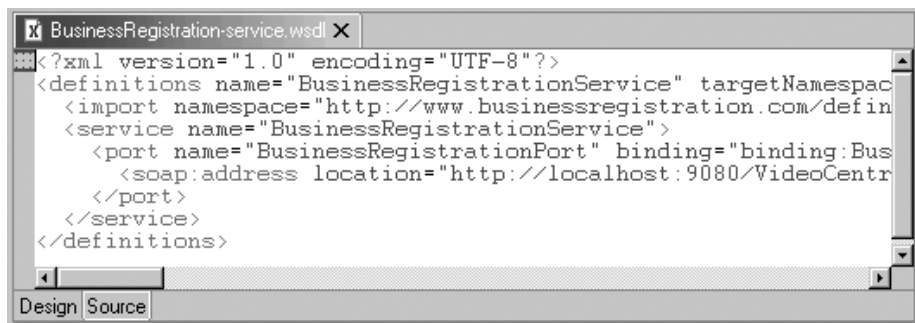
6. Klicken Sie auf **Finish**, um den Service zu implementieren.
7. Wählen Sie die Anzeige 'Servers' in der Serverperspektive aus. Klicken Sie in dieser Anzeige mit der rechten Maustaste auf VideoCentralServer, und wählen Sie anschließend **Stop** aus.

Anmerkung: Wenn die Implementierung fehlgeschlagen ist, finden Sie weitere Informationen zu den möglichen Ursachen in "Kapitel 8". Beheben allgemeiner Fehler".

Zu diesem Zeitpunkt wird die WSDL erstellt. Sie wird im Webprojekt VideoCentral angezeigt:

- VideoCentral > webApplication > wsdl > BusinessRegistration-binding.wsdl
- VideoCentral > webApplication > wsdl > BusinessRegistration-service.wsdl

Wenn Sie die WSDL-Datei mit dem Standardeditor anzeigen möchten, klicken Sie doppelt auf die Datei, und wählen Sie dann den Knopf **Source** (vgl. BusinessRegistration-service.wsdl in Abbildung 27) aus. Das Bindungsdokument definiert das Protokoll und das Format für Operationen und Nachrichten, die mit Hilfe eines bestimmten Porttyps (portType) definiert wurden. Das Service-dokument stellt eine Gruppe von zusammengehörigen Endpunkten für Bindungen im Bindungsdokument dar.



```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="BusinessRegistrationService" targetNamespac
<import namespace="http://www.businessregistration.com/defin
<service name="BusinessRegistrationService">
  <port name="BusinessRegistrationPort" binding="binding:Bus
    <soap:address location="http://localhost:9080/VideoCentr
  </port>
</service>
</definitions>
```

Abbildung 27. Inhalt der WSDL-Datei

Wiederholen Sie Task 2, Schritte 1 - 7, für die verbleibenden fünf Web-Services. Für diese Phase von Video Central stehen die folgenden sechs Services zur Verfügung:

- COM/ibm/db2/samples/videocentral/services/BusinessRegistration.java
- COM/ibm/db2/samples/videocentral/services/CustomerInfraction.java
- COM/ibm/db2/samples/videocentral/services/CustomerRegistration.java
- COM/ibm/db2/samples/videocentral/services/CustomerRentedList.java
- COM/ibm/db2/samples/videocentral/services/WishList.java
- COM/ibm/db2/samples/videocentral/services/MovieSearch.java

Nach der Implementierung aller Web-Services werden zwei WSDL-Dateien für jeden Service in Ihrem Webprojekt (vgl. Abbildung 28) angezeigt.

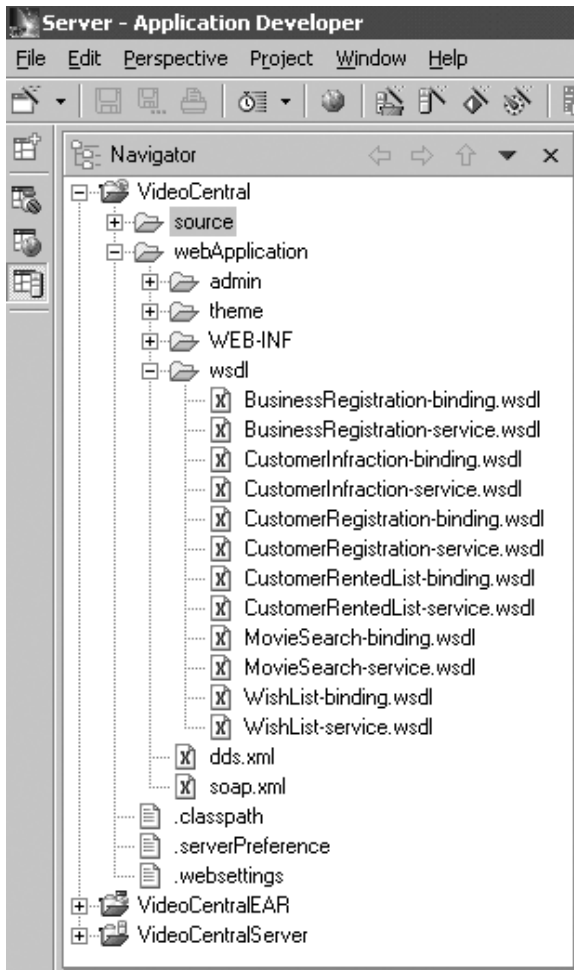


Abbildung 28. Alle Web-Services implementiert

Zusammenfassung der Lerneinheit

In diesem Kapitel wurden die folgenden Themen erläutert:

- Prüfen der Umgebung für die Implementierung von Video Central
- Implementieren der sechs Video Central-Web-Services von WebSphere Studio Application Developer in WebSphere Application Server
- Generieren von Proxyklassen für den Zugriff auf Web-Services von Video Central
- Generieren von WSDL-Dateien für jeden Service. Die WSDL-Dateien können in der UDDI-Registrierdatenbank (vgl. Kapitel 2) veröffentlicht oder für einen Service-Requester bereitgestellt werden.

Nächster Schritt

Als Nächstes wird ein webbasierter Beispielclient zum Aufrufen der Video Central-Services erstellt, um die benutzten Web-Services anzuzeigen.

Kapitel 7. Erstellen und Ausführen eines webbasierten Beispielclients für Video Central

Dauer

Zum Durcharbeiten der vorliegenden Lerneinheit werden ca. 25 Minuten benötigt.

Voraussetzungen

Die Arbeitsschritte in "Kapitel 4. Einführung in WebSphere Studio Application Developer" und "Kapitel 6. Implementieren von Video Central" müssen erfolgreich abgeschlossen worden sein.

Übersicht

Im vorliegenden Kapitel wird das Design eines webbasierten Beispielclients (Service-Requester) für Video Central beschrieben. Darüber hinaus werden die Arbeitsschritte erläutert, die zum Einsatz der bereitgestellten Web-Services erforderlich sind. Diese Arbeitsschritte umfassen das erneute Generieren der Proxyklassen und das Implementieren des Clients für den WebSphere Application Server. In Kapitel 6 wurden die Proxyklassen und WSDL-Dateien auf der Basis von Java-Beans generiert. In diesem Kapitel werden die Proxyklassen aus den WSDL-Dateien erneut generiert.

Design des Beispielclients von Video Central

Die Arbeitsschritte des nächsten Abschnitts werden auf dem Anwendungsserver des Service-Requesters (Video Client) ausgeführt. Der Client erstellt die Proxyklassen für die Web-Services, die vom Web-Service-Provider (vgl. Abbildung 29) bereitgestellt werden.

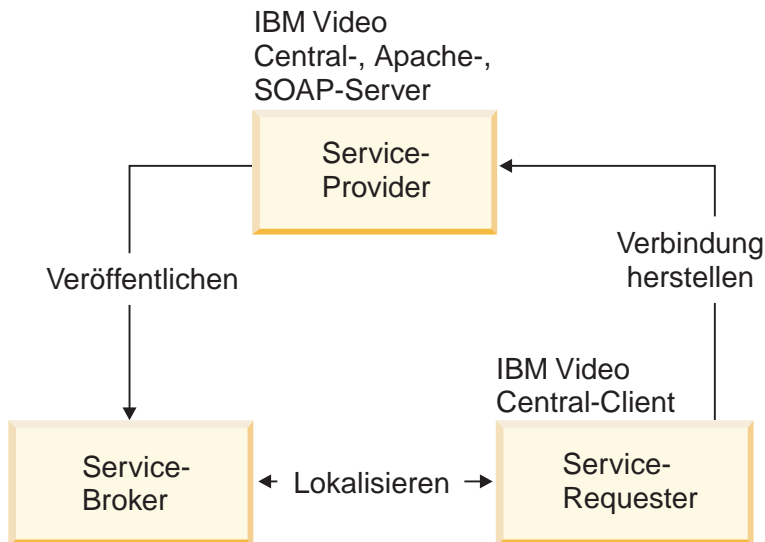


Abbildung 29. Grundlegende Aufgabenbereiche von Web-Services

Bis zu diesem Zeitpunkt wurden alle Dateien auf dem Video Central-Server (Service-Provider) gespeichert. Im nächsten Szenario des vorliegenden Lernprogramms wird der Video Client-Server (Service-Requester) behandelt. Im Rahmen des Lernprogramms wird der gleiche WebSphere Application Server verwendet, in einer realen Systemumgebung wäre Video Client jedoch auf einem anderen Anwendungsserver innerhalb der jeweiligen Videothek gespeichert.

Klassen- und Sequenzdiagramme für den Beispielclient von Video Central

Abbildung 30 zeigt das vollständige Klassendiagramm für den Beispielclient Video Client von Video Central:

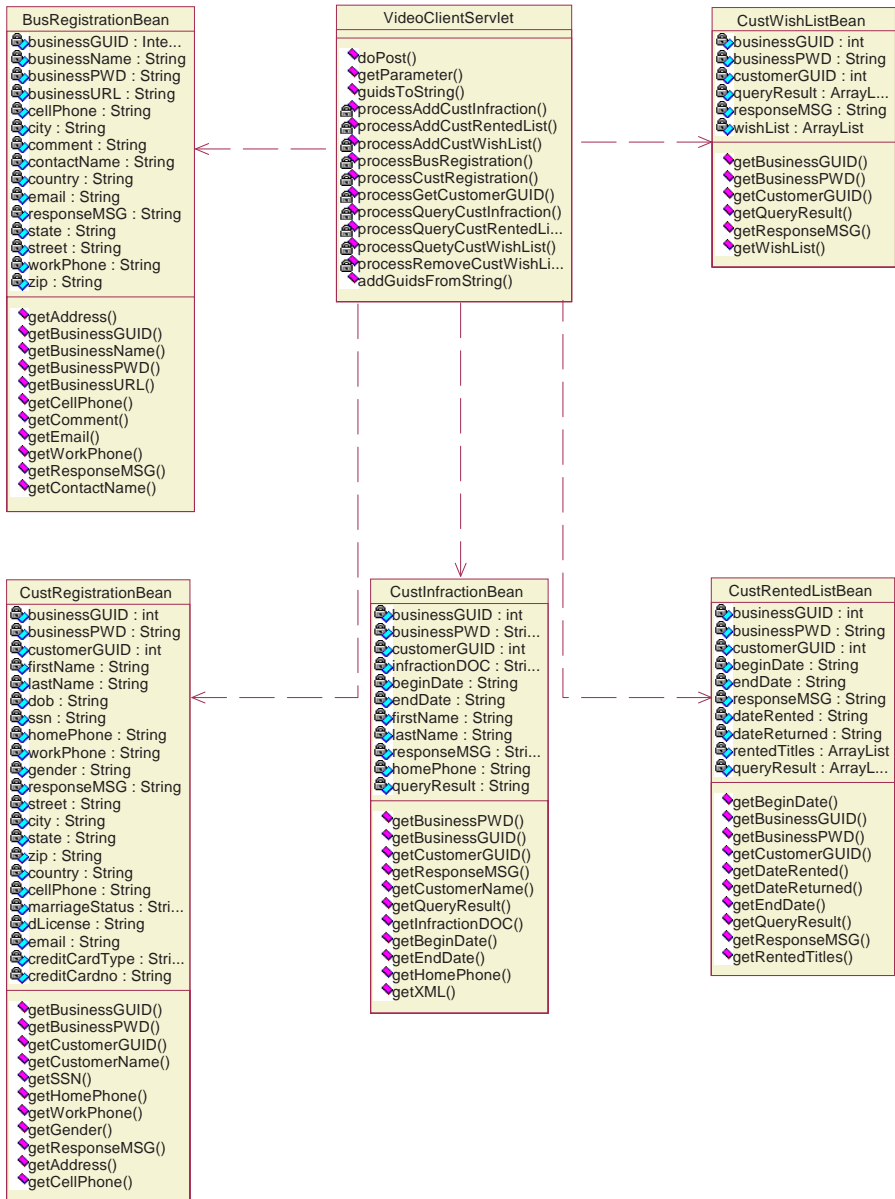


Abbildung 30. Klassendiagramm für den Beispielclient Video Client von Video Central

Abbildung 31 zeigt das vollständige Sequenzdiagramm für den Beispielclient Video Client von Video Central:

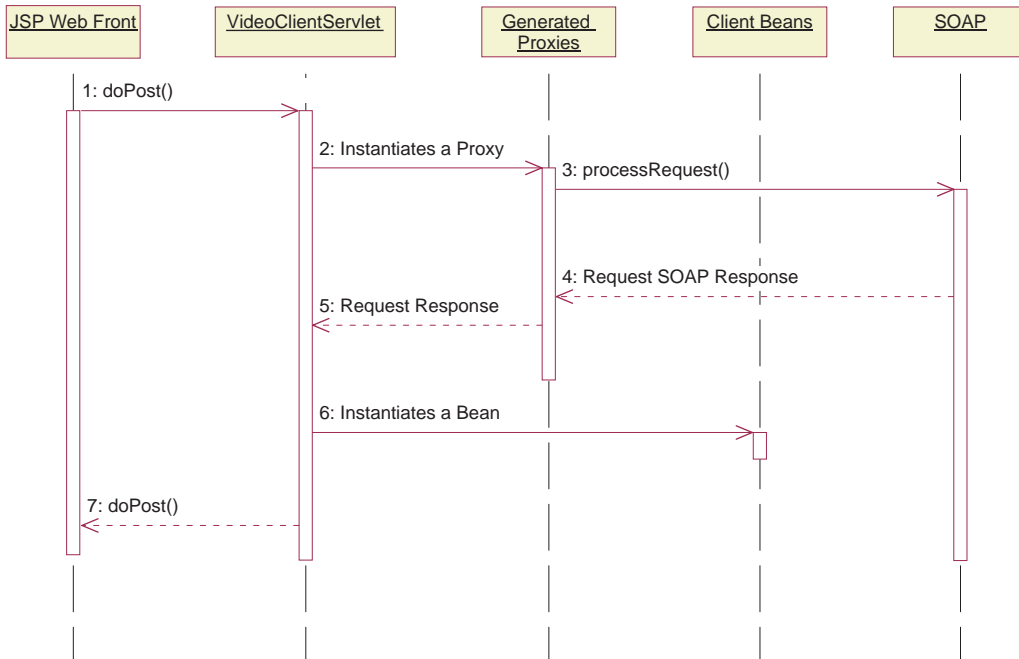


Abbildung 31. Sequenzdiagramm für den Beispielclient Video Client von Video Central

In Video Client werden alle Anforderungen an VideoClientServlet gesendet. Bei der Verarbeitung der Anforderungen ruft VideoClientServlet die entsprechende Proxyanforderung auf, die dann den fernen Web-Service aufruft. Mit den in der Antwort enthaltenen Informationen erstellt VideoClientServlet eine Client-Bean. Abschließend werden die Resultate mit Hilfe von JSP (Java-Server Page) für den Endbenutzer angezeigt.

Video Central-Tasks

Nun werden Sie folgende Arbeitsschritte ausführen:

- Erstellen der erforderlichen Proxyklassen
- Implementieren des Client-Codes für WebSphere Application Server

Task 1: Vorbereiten des Clientprojekts

Führen Sie in WebSphere Studio Application Developer die folgenden Arbeitsschritte aus:

1. Erstellen Sie das Webprojekt VideoClient:
 - a. Heben Sie in der Webperspektive das Webprojekt **VideoCentral** hervor. (Dieses befindet sich innerhalb der Baumstrukturhierarchie von WebSphere Studio Application Developer auf der höchsten Ebene.)
 - b. Erstellen Sie ein Webprojekt, indem Sie auf **File** → **New** → **Web Project** klicken.
 - c. Ordnen Sie dem Webprojekt den Namen VideoClient und dem Unternehmensanwendungsprojekt den Namen VideoClientEAR zu.
 - d. Klicken Sie auf **Next**.
 - e. Daraufhin wird die Anzeige 'Web Settings' angezeigt. Übernehmen Sie alle Standardeinstellungen, und klicken Sie auf **Next**.
 - f. Daraufhin wird die Anzeige **Java Settings** aufgerufen. Wählen Sie die Indexzunge 'Libraries' aus, und klicken Sie auf **Add External JARs**. Fügen Sie im Dialogfenster **JAR Selection** die folgenden externen JAR-Komponenten in der nachstehend aufgeführten Reihenfolge hinzu:
 - <WAS_install_dir>\lib\xerces.jar
 - <WSAD_install_dir>\plugins\com.ibm.etools.webservice\runtime\soap.jar

Die Datei xerces.jar wird vom Web-Service-Requester für Kundenverstöße benötigt, weil Verstöße im Speicher als XML-Objekte bearbeitet werden und xerces ein von IBM bereitgestellter XML DOM-Parser (DOM = Document Object Model) ist.

Jetzt wird die in Abbildung 32 dargestellte Anzeige aufgerufen.

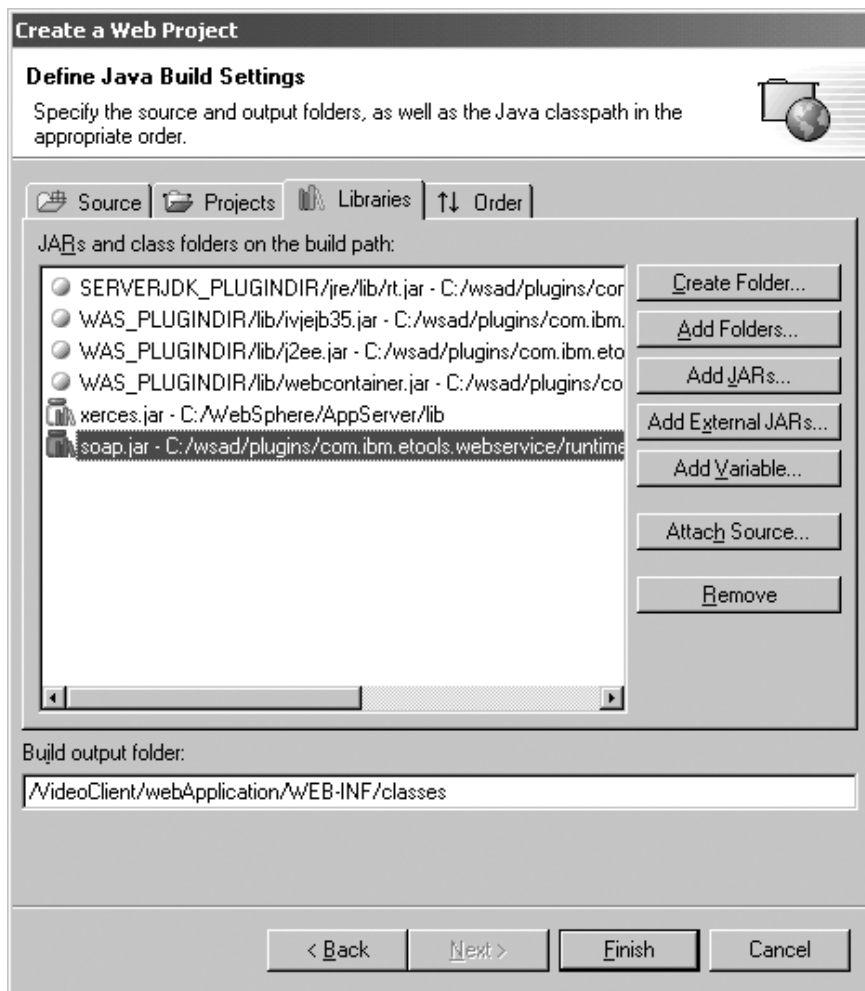


Abbildung 32. Hinzufügen externer JAR-Dateien

g. Klicken Sie auf **Finish**.

2. Fügen Sie VideoClient zum Server hinzu:
 - a. Erweitern Sie in der Anzeige 'Server Configuration' die Anzeige für **Server Configurations**, und klicken Sie mit der rechten Maustaste auf **VideoCentralServer**. Wählen Sie anschließend **Add Project** → **VideoClientEAR** (vgl. Abbildung 33) aus.

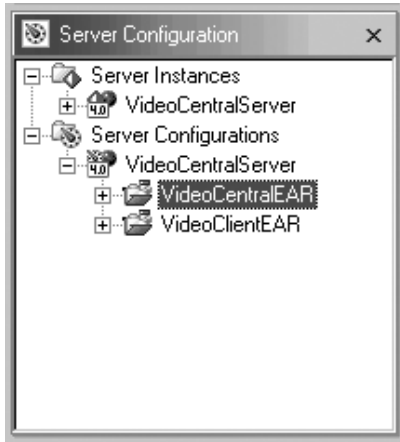


Abbildung 33. Hinzufügen des Client-Web-Projekts zum Server

- b. Klicken Sie in der Anzeige 'Navigator' mit der rechten Maustaste auf das Webprojekt **VideoClient**, und wählen Sie anschließend **Properties** → **Server Preference** aus. Heben Sie im Feld den Eintrag **VideoCentralServer** hervor.
 - c. Klicken Sie auf **Apply** und dann auf **OK**.

3. Importieren Sie den Client-Code in VideoClient:
 - a. Erweitern Sie die Sicht des neu erstellten Webprojekts VideoClient, und heben Sie für das Webprojekt **source** (vgl. Abbildung 34) hervor.

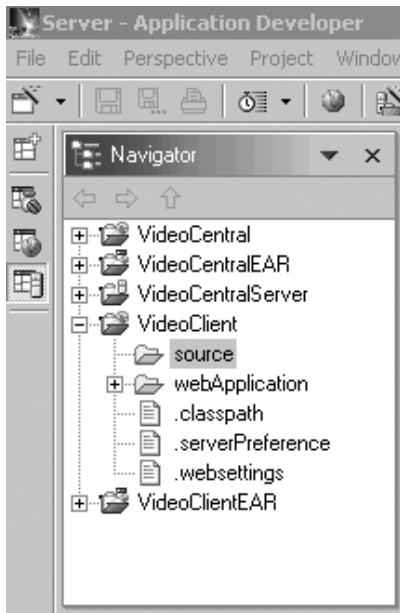


Abbildung 34. Hervorheben von 'source'

- b. Klicken Sie auf **File** → **Import**.
- c. Wählen Sie als Importquelle **Zip file** aus. Klicken Sie auf **Next**.
- d. Suchen Sie nach `<DB2_install_dir>\tutorials\VideoCentral\lib\VideoClient.jar`. Klicken Sie auf **OK**. Daraufhin wird die Anzeige **Zip file** (vgl. Abbildung 35) aufgerufen.

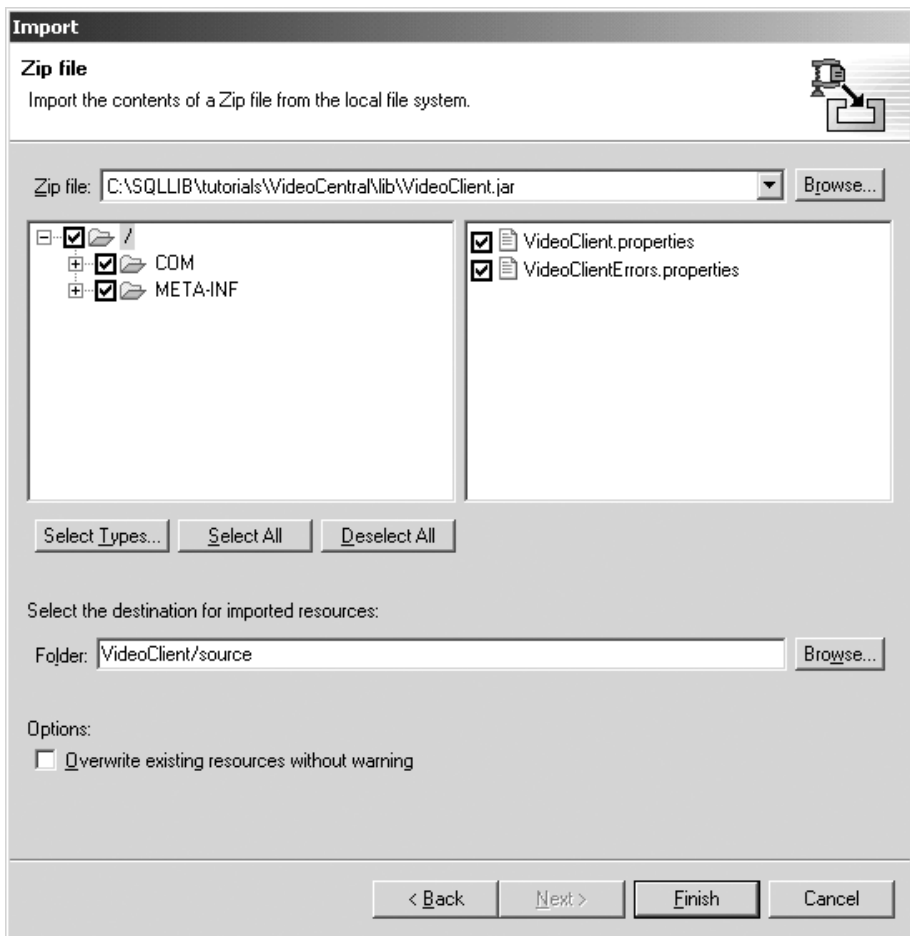


Abbildung 35. Importieren des Client-Codes von einer JAR-Datei in das Projekt VideoClient

- e. Klicken Sie auf **Finish**. Die Dateien werden nun in WebSphere Studio Application Developer importiert und Sie können in der Aktionsleiste sehen, dass die Dateien kompiliert werden.

4. Importieren Sie die Client-JSPs:
 - a. Erweitern Sie die Sicht des Webprojekts VideoClient, und heben Sie für dieses **webApplication** (vgl. Abbildung 36) hervor.

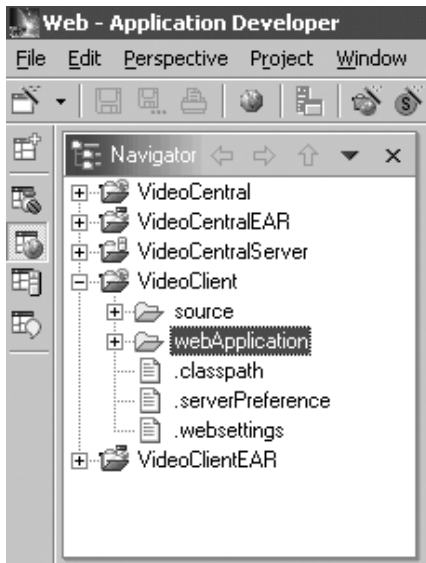


Abbildung 36. Hervorheben von 'webApplication'

- b. Klicken Sie auf **File** —> **Import**.
- c. Wählen Sie als Importquelle **File System** aus. Klicken Sie auf **Next**.
- d. Suchen Sie nach <DB2_install_dir>\tutorials\VideoCentral\jsp, und klicken Sie anschließend auf **OK**. Daraufhin wird die Anzeige 'File system' aufgerufen. Klicken Sie auf den Knopf **Select All** (vgl. Abbildung 37).

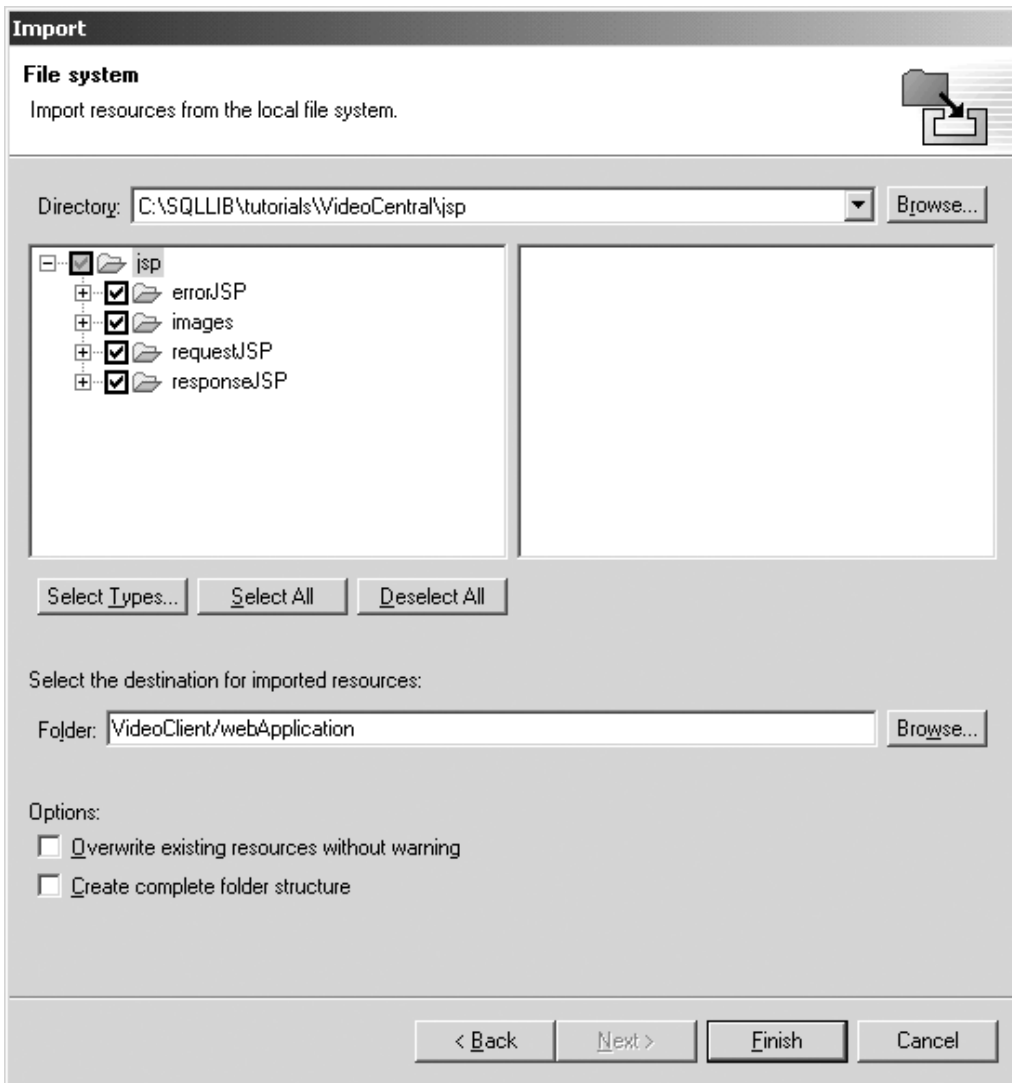


Abbildung 37. Anzeige 'File system'

- e. Klicken Sie auf **Finish**.

Die Dateien werden nun in WebSphere Studio Application Developer importiert und Sie können in der Aktionsleiste sehen, dass die Dateien kompiliert werden.

Nach dem Import der Dateien in das Webprojekt VideoClient müssen Sie über die im Folgenden dargestellte Struktur (vgl. Abbildung 38) verfügen:

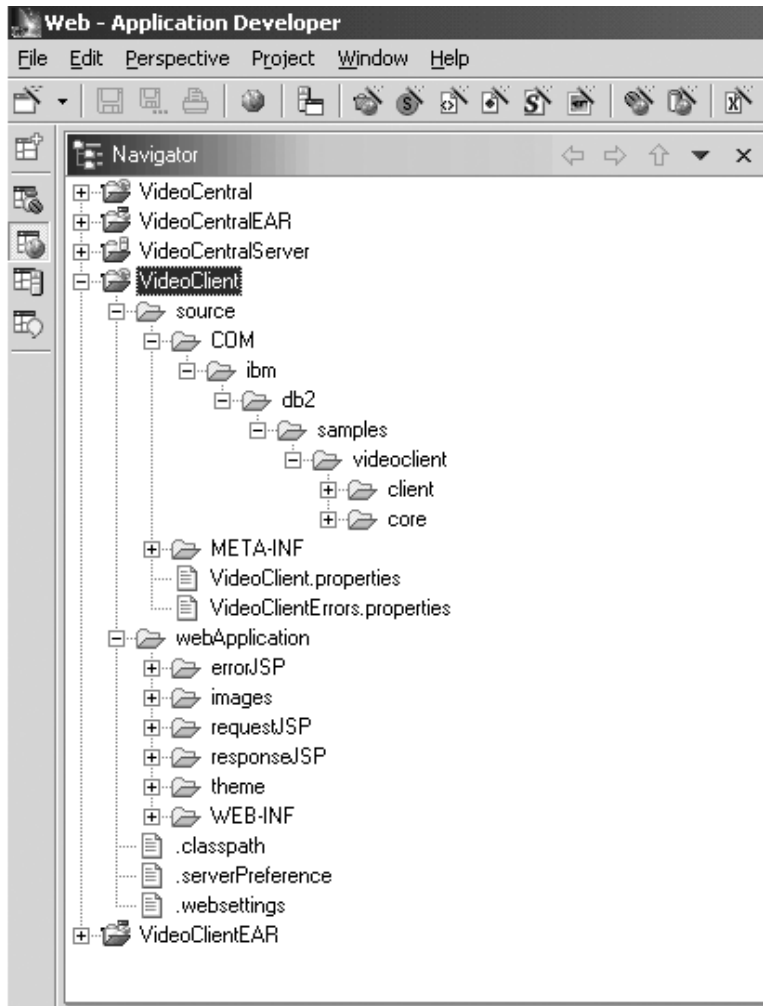


Abbildung 38. Erfolgreicher Import des Video Central-Clients VideoClient

Task 2: Erstellen der Proxies mit WebSphere Studio Application Developer

Zur Ausführung der im folgenden Abschnitt beschriebenen Arbeitsschritte muss Kapitel 4 abgeschlossen worden sein. Zur Ausführung dieser Task werden die im vorherigen Abschnitt erstellten WSDL-Dateien benötigt.

1. Klicken Sie in der Webperspektive im Webprojekt VideoClient mit der rechten Maustaste auf **source/COM/ibm/db2/samples/videoclient**. Wählen Sie dann **New** → **Folder** aus, und erstellen Sie einen neuen Ordner mit dem Namen `proxies`. Im Container muss `VideoClient/source/COM/ibm/db2/samples/videoclient/proxies` angezeigt werden. Klicken Sie auf **Finish**.
2. Generieren Sie den Proxy für den BusinessRegistration-Service.
 - a. Erweitern Sie in der Webperspektive im Webprojekt VideoCentral **webApplication\wsdl**, und klicken Sie mit der rechten Maustaste auf **BusinessRegistration-service.wsdl** → **New** → **Other**. Wählen Sie im daraufhin angezeigten Dialogfenster **Web Services** → **Web Service Client** aus. Klicken Sie auf **Next**.
 - b. Klicken Sie erneut auf **Next**, bis das Dialogfenster **Web Service Binding Proxy Generation** aufgerufen wird.
 - c. Ersetzen Sie im Feld **Folder** den Eintrag `/VideoCentral/source` durch `/VideoClient/source`.
 - d. Ersetzen Sie im Feld **Class** den Eintrag `proxy.soap.BusinessRegistrationProxy` durch Folgendes:
`COM.ibm.db2.samples.videoclient.proxies.BusinessRegistrationProxy`

Anmerkung: Zur Vereinfachung können Sie für die Erstellung der verbleibenden Proxies diesen Pfad in die Zwischenablage kopieren und anschließend als Pfadangabe für die restlichen Proxies einfügen.

Jetzt wird die in Abbildung 39 dargestellte Anzeige aufgerufen.

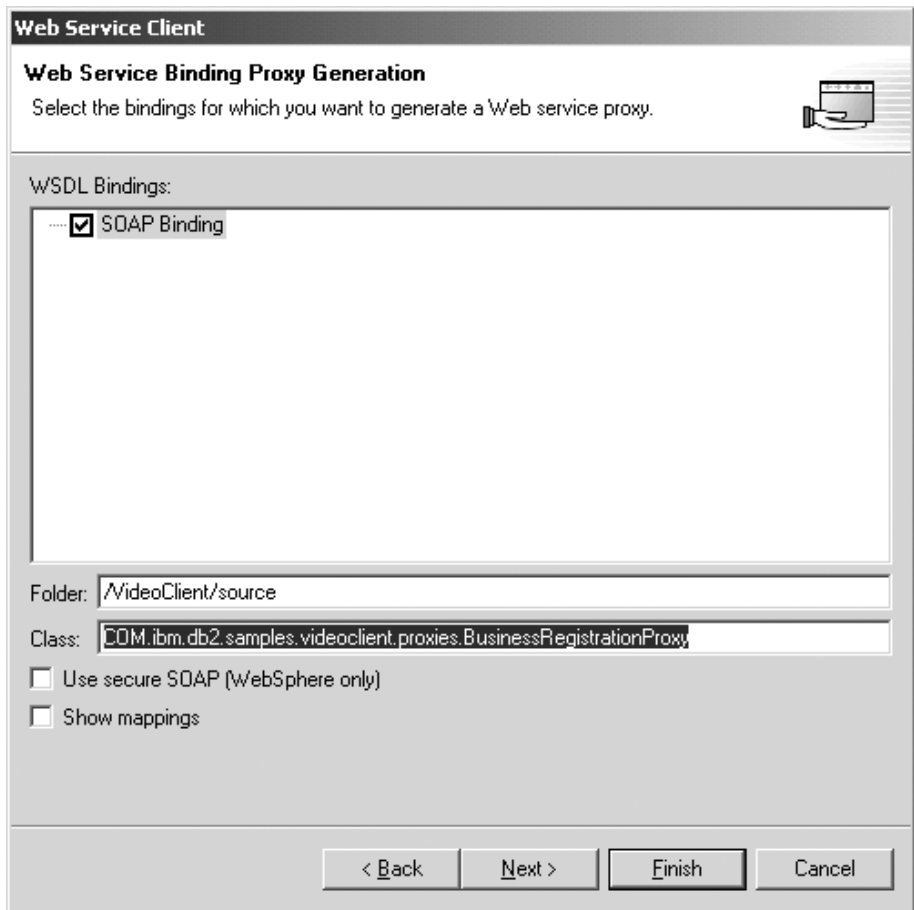


Abbildung 39. Generieren des BusinessRegistration-Proxies

e. Klicken Sie auf **Finish**.

Wiederholen Sie für die verbleibenden fünf WSDL-Dateien **Task 2, Schritt 2**.

- CustomerRegistration-service.wsdl
- CustomerInfracation-service.wsdl
- CustomerRentedList-service.wsdl
- WishList-service.wsdl
- MovieSearch-service.wsdl

Daraufhin werden die Java-Dateien für den Proxy im Webprojekt VideoClient angezeigt. Außer einigen eventuell vorhandenen Warnungen, die jedoch akzeptabel sind (vgl. Abbildung 40) sollten nun keine Compilerfehler mehr in WebSphere Studio Application Developer angezeigt werden.

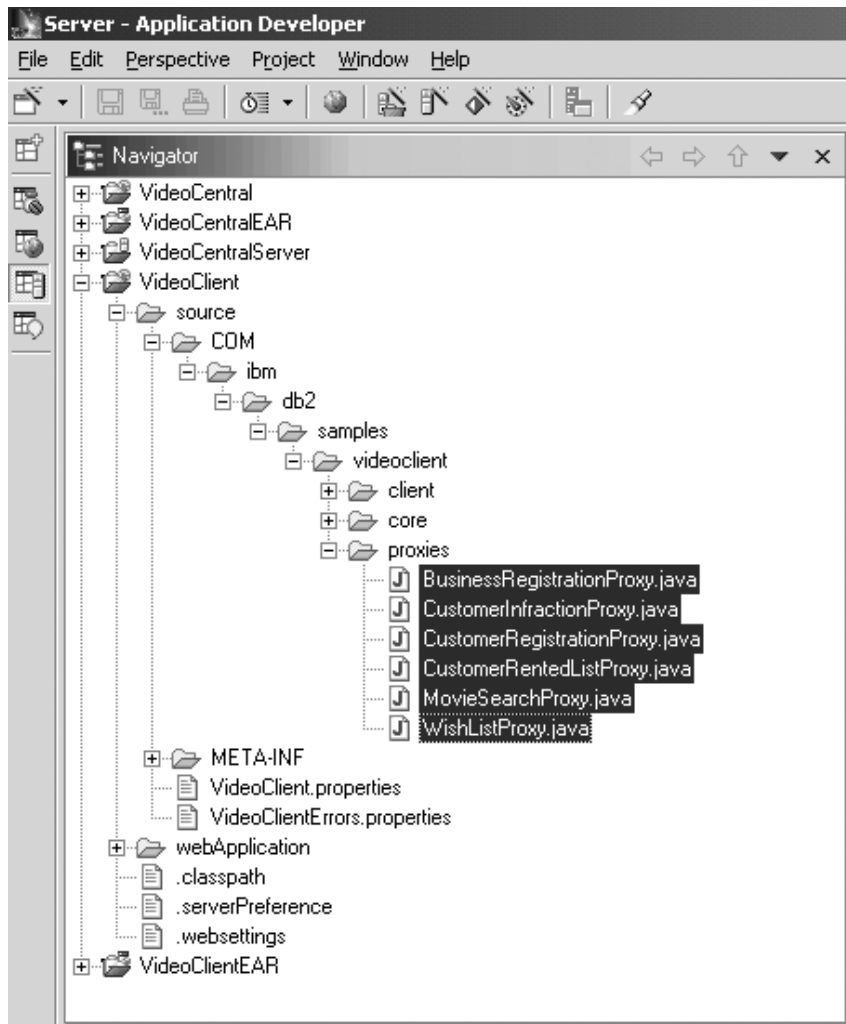


Abbildung 40. Für alle WSDL-Dateien erstellte Proxies

Task 3: Implementieren des Beispielclients im WebSphere Application Server

1. Erweitern Sie in der Webperspektive die Sicht des Ordners **VideoClient\webApplication\WEB-INF**.
2. Öffnen Sie im Ordner **Web-INF** die Datei **web.xml**.
3. Wählen Sie die Indexzunge 'Servlets' aus.
4. Fügen Sie **VideoClientServlet** hinzu:
 - a. Klicken Sie auf den Knopf **Add**, und wählen Sie **VideoClientServlet** aus. Klicken Sie anschließend auf **OK**.
 - b. Heben Sie den Eintrag **VideoClientServlet** hervor, und klicken Sie im Feld **URL mappings** auf **Add**. Geben Sie nun bei (New URL) den Wert **servlet/VideoClient** ein. Jetzt wird die in Abbildung 41 dargestellte Anzeige aufgerufen.

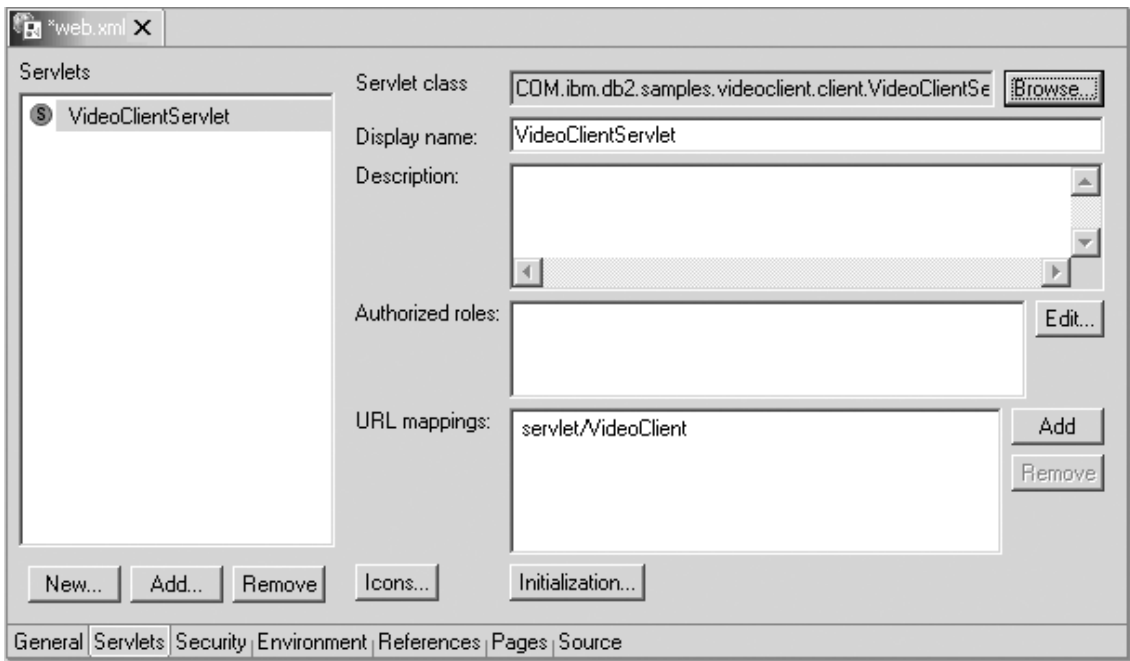


Abbildung 41. Hinzufügen der Client-Servlets

5. Klicken Sie auf **File** → **Save web.xml**.
6. Schließen Sie das Fenster **web.xml**.

Task 4: Ausführen des Beispielclients

1. Um auf den Client zuzugreifen, müssen Sie einen Webbrowser öffnen und dort die folgende Adresse aufrufen:
`http://localhost:9080/VideoClient/requestJSP/VideoClientRequest.jsp`
2. Jetzt können Sie mit dem Client arbeiten. Um mit der Verwendung von Video Central zu beginnen, müssen Sie ein Unternehmen registrieren. Da noch keine Kunden registriert sind, müssen Sie als Nächstes eine Kundenregistrierung ausführen. Zum Hinzufügen von Daten zu einem Kundenverstoß wird DB2 XML Extender verwendet. Anschließend können Sie die Daten zu den gespeicherten Kundenverstößen abfragen. Wenn Sie mit einem Webbrowser arbeiten, der XML unterstützt, wird das XML-Dokument mit den Informationen zu den Kundenverstößen im Browser angezeigt. Jetzt können Sie sich mit den anderen Services vertraut machen.

Anmerkung: Die Datei `info.vc` wird im Verzeichnis `c:\` Ihres Computers erstellt. Im aktuellen Design von Video Central wird diese Datei gelesen und geschrieben, um die Kundenunternehmen zu registrieren. Dieser Ansatz ist nicht ganz optimal und wird in zukünftigen Produktversionen geändert werden. Da es sich bei Video Central um eine B2B-Anwendung handelt, steht zwischen dem Aufruf der Web-Services kein Kontext zur Verfügung. Mit Hilfe der Datei `info.vc` konnten Anforderungen bestimmter Unternehmen (Kunden) ausgeführt werden, ohne dass eine ID und das zugehörige Kennwort für die Anforderung abgefragt werden musste. Im Idealfall wird für jede der Web-Service-Anforderungen ein anerkanntes Sicherheitsberechtigungstoken codiert.

Zusammenfassung der Lerneinheit

In diesem Kapitel wurden die folgenden Themen erläutert:

- Übersicht zum Design des Video Central-Beispielclients
- Erstellen der Proxyklassen
- Implementieren des Client-Codes
- Ausführen des Clients

Kapitel 8. Beheben allgemeiner Fehler

1. Wenn das Implementieren von Code von WebSphere Studio Application Developer in WebSphere Application Server fehlschlägt, müssen Sie prüfen, ob der WebSphere-Installationspfad korrekt ist. Die Standardeinstellung in der Anzeige **Java bean Web Service Deployment** von WebSphere Studio Application Developer lautet C:\WebSphere\AppServer. Ändern Sie diese Angabe in das korrekte Installationsverzeichnis für WebSphere Application Server in Ihrer Systemumgebung.
2. Wird die Fehlermeldung "Launching the server failed: Trace service port 7000 is in use. ORB bootstrap port 900 is in use. Change each used port number to another unused port on the ports page of the server configuration editor. In case you have another WebSphere server running you can try to increase each used port number by one and try again" angezeigt, müssen Sie die folgenden Arbeitsschritte ausführen:

Wählen Sie in WebSphere Studio Application Developer die Anzeige 'Servers' in der Serverperspektive aus. Klicken Sie in dieser Anzeige mit der rechten Maustaste auf **VideoCentralServer**, und wählen Sie anschließend **Stop** aus.
3. Wenn Sie Fehlermeldungen erhalten, während Sie über den Browser auf die von Video Central bereitgestellten Web-Services zugreifen, müssen Sie als Erstes prüfen, ob der Serverstatus des Serverexemplars "Server is synchronized" lautet. Möglicherweise müssen die verfügbaren Webprojekte in WebSphere Studio Application Developer erneut veröffentlicht werden.
4. Wird die Fehlermeldung "SOAPException: faultCode=SOAP-ENV:Client; msg=Error opening socket: Connection refused: no further information; targetException=java.lang.IllegalArgumentException: Error opening socket: Connection refused: no further information]" angezeigt, während Sie auf die Web-Services von Video Central zugreifen, müssen Sie prüfen, ob die Portnummer, die von der entsprechenden Proxyklasse zur Kommunikation mit dem Service verwendet wird, mit der Portnummer übereinstimmt, an der die SOAP-Laufzeitkomponente empfangsbereit ist (im vorliegenden Lernprogramm lautet diese Portnummer 9080).

5. Wird die Fehlermeldung "DES7300N Index has not been activated, SQL0438N" ausgegeben, müssen Sie den Eintrag für die Position der Net Search-Indizes (mvnets.inxdir) in der Datei VideoCentral.properties prüfen, um sicherzustellen, dass die Pfadangabe korrekt ist. Darüber hinaus kann es erforderlich sein, die folgenden Befehle auszuführen:
 - db2nx update index name database sample
 - db2nx update index plot database sample
6. Zur Behebung von Anwendungsfehlern in WebSphere Studio Application Developer stehen im Serverkonsolenprotokoll in der Anzeige 'Servers' wichtige Informationen zur Fehlerbehebung zur Verfügung.

Kapitel 9. Zusätzliche Informationen

- IBM DB2: <http://www.ibm.com/software/data/db2/>
- XML Extender-Dokumentation:
<http://www.ibm.com/software/data/db2/extenders/xmlext/library.html>
- Net Search Extender-Dokumentation:
<http://www.ibm.com/software/data/db2/extenders/netsearch/download.html>
- SOAP: <http://www.w3.org/TR/SOAP/>
- Java-Programmierung: <http://www.javasoft.com>
- WSDL: <http://www.w3.org/TR/wsdl>
- Web-Service-Informationen:
<http://www.ibm.com/developerworks/webservices/>
- IBM DeveloperWorks - XML: <http://www.ibm.com/developerworks/xml/>
- WebSphere-Entwicklerdomäne: <http://www7b.boulder.ibm.com/wsdd/>
- DB2-Entwicklerdomäne: <http://www7b.boulder.ibm.com/dmdd/>
- DB2-Web-Services: <http://www.ibm.com/software/data/webservices/>
- WebSphere/DB2-Integration:
<http://www.ibm.com/software/data/integration/WebSphere>

Anhang A. Referenzinformationen

DB2 XML Extender - Produktbeschreibung

DB2 XML Extender ermöglicht Ihnen die Integration der Leistungsfähigkeit von IBM DB2 Universal Database mit der Flexibilität von XML. Das Produkt unterstützt die Speicherung von und den Zugriff auf XML-Dokumente, das Generieren von XML-Dokumenten auf der Basis vorhandener relationaler Daten und das Aufgliedern von XML-Dokumenten in relationale Daten. (Hierbei werden die Daten zerlegt und ohne Befehle und sonstige Steuerelemente gespeichert.)

DB2 XML Extender stellt neue Datentypen, Funktionen und gespeicherte Prozeduren zur Verfügung, um XML-Daten unter DB2 zu verwalten. Darüber hinaus ist es möglich, XML-Daten zu zerlegen und die Einzelkomponenten als Spalten in mehreren Tabellen zu speichern. Zum schnelleren Abrufen können für die Elemente oder Attribute eines XML-Dokumentes auch Indizes definiert werden. Über Text-Extender können auch Funktionen für die Text- und Abschnittssuche in den XML-Spalten oder den Einzelkomponenten verwendet werden.

Für den Datenaustausch in einer Großhandlungsumgebung (B2B) können auch XML-Dokumente auf der Basis vorhandener DB2-Tabellen definiert werden. Net.Data und DB2 XML Extender können zum Generieren von XML-Dokumenten über DB2 benutzt werden. Diese Dokumente können anschließend an Konsumenten verteilt und von diesen mit einem Browser angezeigt werden.

DB2 XML Extender stellt die folgenden Funktionen zur Verfügung, mit denen Sie XML-Daten in DB2 verwalten und benutzen können:

- Verwaltungstools zur Integration von XML-Daten in relationalen Tabellen.
- Speicher- und Verwendungsmethoden für XML-Daten: XML-Spalten, XML-Objektgruppen.
- DTD-Repository (DTD = Document Type Definition) mit der Bezeichnung DTD_REF, in dem DTD-Komponenten zum Prüfen von XML-Daten gespeichert werden können.
- Ein als DAD-Datei (DAD = Document Access Definition) bezeichnetes Zuordnungsschema, mit dem XML-Dokumente den entsprechenden relationalen Daten zugeordnet werden können. DAD-Dateien werden mit Hilfe der Tabelle XML_USAGE verwaltet, die erstellt wird, wenn Sie die XML-Unterstützung der Datenbank aktivieren.
- Leistungsstarke benutzerdefinierte Funktionen (UDFs = User-Defined Functions) zum Speichern und Abrufen von XML-Dokumenten in XML-Spalten

sowie zum Extrahieren von XML-Elementen oder -Attributwerten. Bei einer UDF handelt es sich um eine Funktion, die im Datenbankverwaltungssystem definiert wird und anschließend in SQL-Abfragen eingesetzt werden kann. DB2 XML Extender stellt die folgenden UDF-Typen zur Verfügung:

- Speichern: Zur Speicherung intakter XML-Dokumente in XML-Spalten als XML-Datentypen.
- Extrahieren: Zum Extrahieren von XML-Dokumenten oder der angegebenen Elemente und Attribute als grundlegende Datentypen.
- Aktualisieren: Zum Aktualisieren ganzer XML-Dokumente oder angegebener Elemente und Attributwerte.

Alle UDFs des DB2 XML Extender weisen das Präfix `db2xml` auf, bei dem es sich um den Schemennamen der benutzerdefinierten Funktionen von DB2 XML Extender handelt.

DB2 Net Search Extender - Produktbeschreibung

Mit dem DB2 Net Search Extender können Sie in DB2 Universal Database integrierte Net.Data-, Java- oder DB2-CLI-Anwendungen durch eine schnelle und leistungsstarke Volltextabruffunktion ergänzen.

DB2 Net Search Extender ermöglicht das schnelle Indexieren umfangreicher Datenvolumen, die Hochgeschwindigkeitssuche mit einer großen Anzahl gleichzeitiger Benutzer sowie die Speicherung vorsortierter Tabellenspalten im Hauptspeicher während der Indexierung. Hierdurch kann bei Suchoperationen der kostenintensive Datenbankzugriff und die Anzahl der Pagingoperationen reduziert werden.

DB2 Net Search Extender umfasst die folgenden Funktionen, die die Funktionalität von DB2 Text Extender und DB2 Text Information Extender ergänzen:

- Indexierung:
 - Ein sehr schneller Indextyp (Ngram).
 - Möglichkeit zur Verwendung mehrerer Indizes in einer Textspalte.
 - Indexierung ohne Datensperrung.
 - Dynamische Indexaktualisierung zum Nachvollziehen von Datenbankänderungen.
- Suche:
 - Bereitstellung einer gespeicherten Prozedur an Stelle von UDFs.
 - Unterstützung von Suchoperationen auf Wort-, Satz- und Wortstamm-basis sowie Suche nach grober Übereinstimmung.
 - Identifikation und Einschränkung von Suchoperationen auf Abschnitte in mit spezifischen Befehlen markierten Dokumenten.

- Unterstützung numerischer Suchoperationen für bestimmte Wertebereiche.
- Unterstützung Boolescher Suchoperationen und Suchoperationen mit Platzhalterzeichen.
- Suchergebnis:
 - Möglichkeit zur Angabe der Sortierfolge der Suchergebnisse während der Indexierung oder Verwendung benutzerdefinierter Rangordnungsbereiche für die Sortierung.
 - Möglichkeit zur Angabe von Suchergebnisuntergruppen bei Suchoperationen in umfangreichen Datenbeständen, für die sehr große Ergebnislisten zu erwarten sind.
 - Möglichkeit zur Angabe von Einschränkungen für die Suchoperation bei Suchbegriffen mit einer hohen Trefferrate.
 - Möglichkeit zur Positionierung des Cursors (Cursoreinstellung) auf den Suchergebnissen.

IBM WebSphere Studio Application Developer - Produktbeschreibung

Bei IBM WebSphere Studio Application Developer (WSAD) handelt es sich um das Nachfolgeprodukt zu WebSphere Studio Professional und WebSphere Studio Advanced Edition sowie VisualAge für Java Enterprise Edition. Diese Produkte unterstützen die vollständige Entwicklung sowie das Testen und Implementieren abgeschlossener e-business-Anwendungen.

Die neuen WebSphere Studio-Produkte verfügen über ein grundlegend neues Design, um die individuellen Anforderungen aller neuen Anwendungsarten vollständig zu erfüllen. Diese Anforderungen umfassen die Implementierung offener Standards, die Unterstützung von Java, XML und Web-Services sowie Testfunktionen und unterschiedliche Integrationsstufen mit anderen Komponenten und ISV-Produkten. Darüber hinaus ist die Unterstützung von Plug-in-Funktionen, Erweiterungsmöglichkeiten und der aufgabenbereichsbasierten Entwicklung sowie einer verbesserten Benutzerfreundlichkeit und Teamfähigkeit und kürzerer Zeitzyklen bis zur Marktreife erforderlich. WebSphere Studio bietet integrierte Entwicklungstools für alle e-business-Entwicklungsbereiche von Webentwicklern bis hin zu Java-Entwicklern, von Unternehmensanalytikern bis zu Architekturspezialisten und Unternehmensprogrammierern.

WebSphere Studio kombiniert die wichtigsten Funktionen der "klassischen" WebSphere Studio-Produkte und von VisualAge für Java mit den Vorteilen der neuesten Technologien. Auf diese Weise wird die Unterstützung offener Standards, die Integration von Tools, eine verbesserte Flexibilität und die Möglichkeit zur Integration vorhandener Anwendungen geboten.

Informationen zur Web-Services-Umgebung

WebSphere Studio umfasst Assistenten und andere Tools, die zur schnellen Entwicklung von Web-Services beitragen.

Die Entwicklungstools für Web-Services in WebSphere Studio basieren auf offenen, plattformübergreifenden Standards:

- ***Universal Description Discovery and Integration (UDDI)***: Diese Funktion ermöglicht Unternehmen über sog. UDDI-Onlineregistrierdatenbanken die Beschreibung eigener Prozesse, das Veröffentlichen technischer Spezifikationen zur Ausführung von e-business-Operationen mit anderen Unternehmen und die Suche nach anderen Unternehmen, die benötigte Waren und Dienstleistungen anbieten.
- ***Simple Object Access Protocol (SOAP)***: Hierbei handelt es sich um einen Standard für die zuverlässige Übertragung elektronischer Unternehmensnachrichten von einer Unternehmensanwendung zu einer anderen über das Internet.

SOAP gehört zu den Lightweight-Protokollen für den Informationsaustausch in einer dezentralen, verteilten Systemumgebung. Es ist ein auf XML basierendes Protokoll, das sich aus drei Komponenten zusammensetzt: eine Oberfläche, die den Rahmen zur Beschreibung von Nachrichteninhalten und Anweisungen zu deren Verarbeitung definiert, eine Reihe von Codierungsregeln zur Angabe von Exemplaren anwendungsdefinierter Datentypen und eine Konvention zur Darstellung ferner Prozeduraufrufe und Antworten.

SOAP kann zusammen mit einer Vielzahl anderer Protokolle eingesetzt werden. Im vorliegenden Dokument werden jedoch nur die Verbindungen zwischen SOAP und HTTP bzw. HTTP Extension Framework erläutert.

Informationen zur Spezifikation für SOAP Version 1.1 und zusätzliche SOAP-Referenzinformationen finden Sie unter folgender Adresse:

<http://www.w3.org/TR/SOAP/>

- ***Web Services Description Language (WSDL)***: Mit dieser Sprache können Programme beschrieben werden, auf die über das Internet (oder andere Netzwerke) zugegriffen werden kann. Darüber hinaus ermöglicht WSDL das Definieren der Nachrichtenformate und Protokolle für den Datenaustausch mit diesen Programmen.

WebSphere Studio ermöglicht die Ausführung der folgenden Prozesse, die zur Erstellung und Implementierung von Anwendungen mit Web-Service-Unterstützung dienen:

- **Discovery-Unterstützung**. Durchsuchen der UDDI-Unternehmensregistrierdatenbank zur Feststellung bereits vorhandener Web-Services für die Integration.

- **Erstellung oder Umwandlung.** Erstellen von Web-Services aus vorhandenen Komponenten wie z. B. Beans, URLs zur Ein- und Rückgabe von Daten, DB2 XML Extender-Aufrufe, gespeicherte DB2-Prozeduren und SQL-Abfragen.
- **Build.** Aufbau vorhandener Komponenten zu SOAP- und HTTP-Services mit GET/POST-Unterstützung und Beschreibung dieser Komponenten in WSDL. Die Assistenten für Web-Services unterstützen Sie beim Generieren eines SOAP-Proxies für in WSDL beschriebene Web-Services und beim Generieren von Beangerüsten mit Hilfe von WSDL.
- **Implementierung.** Implementieren von Web-Services in WebSphere Application Server- oder Tomcat-Testumgebungen mit Hilfe von Server-Tools.
- **Test.** Testen von lokal oder fern ausgeführten Web-Services zum Abrufen direkter Rückmeldungen.
- **Entwicklung.** Generieren von Beispielanwendungen zur Erstellung eigener Clientanwendungen für Web-Services.
- **Veröffentlichung.** Veröffentlichen von Web-Services in der UDDI-Unternehmensregistrierdatenbank und Werben für eigene Web-Services, so dass diese von anderen Unternehmen genutzt werden können.

Weitere Informationen zu Web-Service-Tools finden Sie in der Onlinehilfe, die bei der Installation von WebSphere Studio Application Developer implementiert wird.

Informationen zur XML-Entwicklungsumgebung

WebSphere Studio stellt eine umfassende XML-Entwicklungsumgebung bereit, die Tools zum Erstellen von DTDs, XML-Schemata und -Dateien umfasst. Darüber hinaus unterstützt das Produkt die Integration relationaler Daten und XML-Komponenten.

Die folgenden XML-Tools stehen zur Verfügung:

Der XML-Editor ist ein Tool zum Erstellen, Anzeigen und Prüfen von XML-Dateien. Sie können ihn zum Erstellen neuer XML-Dateien ohne Vorlage oder auf der Basis vorhandener DTDs oder XML-Schemata verwenden. Darüber hinaus kann er zum Editieren von XML-Dateien, zur Zuordnung dieser Dateien zu DTDs oder Schemata und zu Prüfzwecken benutzt werden.

Der DTD-Editor ist ein Tool zum Erstellen, Anzeigen und Prüfen von DTDs. Mit Hilfe dieses Editors können Sie DTD-Elemente, -Attribute, -Entitäten und -Notationen erstellen und prüfen. Sie können XML-Schemadateien und Java-Beans für die Erstellung von XML-Exemplaren eines XML-Schemas generieren. Darüber hinaus ist es möglich, den DTD-Editor zum Generieren eines HTML-Standardformulars auf der Basis der erstellten DTDs zu verwenden.

Der XML-Schemaeditor ermöglicht Ihnen das Erstellen, Anzeigen und Prüfen von XML-Schemata. Sie können diesen Editor zur Ausführung von Tasks wie z. B. zum Erstellen von XML-Schemakomponenten, Importieren und Anzeigen von XML-Schemata, Generieren von DTDs und Definitionen relationaler Tabellen aus XML-Schemata, Generieren von Java-Beans zur Erstellung von XML-Exemplaren eines XML-Schemas und zum Generieren von DDL-Komponenten aus einem XML-Schema verwenden.

Der XSL-Traceeditor ermöglicht Ihnen die Anwendung einer XSL-Formatvorlage für ein XML-Dokument. Auf diese Weise können Sie ein Ergebnisdokument (HTML oder XML) erstellen. Sie können XML-Dokumente in HTML-, Text- oder andere XML-Dokumenttypen umsetzen. Der Editor zeigt die drei Dokumente (Ergebnis, XML-Quelle und XSL-Quelle) an und ermöglicht Ihnen die visuelle Prüfung des XSL-Umsetzungsscripts und der Beziehungen zwischen den drei Dokumenten.

Der XML-zu-XML-Zuordnungeditor dient zum Zuordnen eines oder mehrerer XML-Quelldokumente zu einem einzigen XML-Zieldokument. Sie können eine Quellendatei (DTD- oder XML-Format) und eine Zieldatei angeben und anschließend die Zuordnungen zwischen Quellen- und Zielkomponenten definieren. Jede Zuordnung umfasst die Auswahl eines Zielfeldes, einer Konvertierungsfunktion sowie der entsprechenden Quellenfelder. Die Zuordnungen können editiert, gelöscht oder zur späteren Verwendung gespeichert werden. Nach dem Definieren der Zuordnungen können Sie ein XSLT-Script generieren, das dann zum Kombinieren und Umsetzen aller XML-Dokumente verwendet werden kann, die den Einstellungen der DTD-Quellenkomponenten entsprechen.

Der XML- und SQL-Abfrageassistent wird zum Erstellen einer XML-Datei auf der Basis der Ergebnisse einer SQL-Abfrage verwendet. Sie können optional ein XML-Schema oder eine DTD-Datei erstellen, in der die Struktur der XML-Datei definiert wird, so dass sie in anderen Anwendungen eingesetzt werden kann. Darüber hinaus können Sie den XML- und SQL-Abfrageassistenten auch zum Erstellen einer DADX-Datei verwenden, die zusammen mit dem Web-Service-Tool verwendet werden kann. Die generierte DADX-Datei enthält dann die SQL-Abfrage.

Der RDB-zu-XML-Zuordnungseditor vereinfacht das Definieren der Zuordnungen zwischen relationalen Tabellen und einer DTD-Datei. Sie können Zuordnungen von Spalten in einer oder mehreren relationalen Tabellen zu Elementen und Attributen in einem XML-Dokument vornehmen. Sie können ein DAD-Script (DAD = Document Access Definition) generieren, das von IBM DB2 Extender zum Erstellen von XML-Dokumenten aus vorhandenen DB2-Daten oder zum Zerlegen von XML-Dokumenten in DB2-Daten verwendet wird. Darüber hinaus können Sie auch Testumgebungen zum Testen der generierten DAD-Dateien erstellen.

Weitere Informationen zu XML-Tools finden Sie in der Onlinehilfe, die bei der Installation von WebSphere Studio Application Developer implementiert wird.

Anhang B. Glossar

DAD - Document Access Definition. Diese Komponente wird zum Definieren des Indexierungsschemas für XML-Spalten oder zum Definieren eines Zuordnungsschemas einer XML-Objektgruppe verwendet.

DTD - Document Type Definition. Bei einer DTD-Komponente handelt es sich um eine Datei (bzw. mehrere gemeinsam verwendete Dateien), die in der XML-Deklarationssyntax geschrieben wurden und eine formale Beschreibung eines bestimmten Dokumenttyps enthält. In dieser Datei wird definiert, welche Namen für Elementtypen verwendet werden, wo diese auftreten und wie sie zusammen eingesetzt werden können.

JDBC - Java Database Connectivity. Eine Anwendungsprogrammierschnittstelle (API), die die gleichen Merkmale aufweist wie ODBC (Open Database Connectivity), die jedoch speziell zur Verwendung durch Java-Datenbankanwendungen konzipiert wurde.

SOAP - Simple Open Access Protocol. Eine Protokollspezifikation, die ein einheitliches Verfahren zur Übergabe von XML-Daten definiert.

UDDI - Universal Description, Discovery and Integration Service. Ein Registrierdatenbank-Mechanismus, der Kunden das dynamische Suchen nach anderen Web-Services ermöglicht.

WSDL - Web Services Description Language. Ein XML-Format, das die Funktionsweise einer Web-Services, seine Speicherposition und das zugehörige Aufrufverfahren beschreibt.

XML - eXtensible Markup Language. Ein universelles Format für strukturierte Dokumente und Daten im Web. Es ist erweiterbar, da es kein festes Format wie z. B. HTML aufweist und stellt eine Untergruppe von SGML (Standard Generalized Markup Language) dar.

XSL - eXtensible Style Language. Definiert die Standardsprache für Formatvorlagen von XML.

Anhang C. Bemerkungen

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation.

- DB2
- IBM
- VisualAge
- WebSphere

Microsoft und Windows NT sind in gewissen Ländern eingetragene Marken der Microsoft Corporation.

Java und alle auf Java basierenden Marken und Logos sind in gewissen Ländern Marken oder eingetragene Marken von Sun Microsystems, Inc.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken oder Dienstleistungsmarken anderer Unternehmen sein.

Kontaktaufnahme mit IBM

Bei technischen Problemen lesen Sie bitte die Informationen zu den Korrekturmaßnahmen, die in "Kapitel 8. Beheben allgemeiner Fehler", und im DB2-Handbuch zur Fehlerbehebung aufgeführt sind, und führen Sie diese aus, bevor Sie sich mit der IBM Kundenunterstützung in Verbindung setzen. Im DB2-Handbuch zur Fehlerbehebung sind die Informationen aufgeführt, die Sie für die DB2-Kundenunterstützung bereithalten sollten, um einen optimalen Service zu gewährleisten.

Wenn Sie weitere Informationen benötigen oder eines der DB2 Universal Database-Produkte bestellen möchten, setzen Sie sich mit einem IBM Ansprechpartner in einer lokalen Geschäftsstelle oder einem IBM Software-Vertriebspartner in Verbindung.

Telefonische Unterstützung erhalten Sie unter den folgenden Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0190/772 243 erreichen Sie die DB2 Helpline, wo Sie Antworten zu DB2-spezifischen Problemen erhalten.

Produktinformationen

Informationen zu DB2 Universal Database-Produkten erhalten Sie telefonisch oder im World Wide Web unter <http://www.ibm.com/software/data/db2/udb/winos2unix/support>.

Diese Site enthält die neuesten Informationen zur technischen Bibliothek, zum Bestellen von Büchern, zu Client-Downloads, Newsgroups, FixPaks, Neuerungen und Links auf verfügbare Webressourcen.

Telefonische Unterstützung erhalten Sie unter den folgenden Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180/55 090 können Sie Handbücher telefonisch bestellen.

Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können, finden Sie auf der globalen IBM Internet-Seite unter folgender Adresse: <http://www.ibm.com/planetwide>

In einigen Ländern sollten sich die IBM Vertragshändler an die innerhalb ihrer Händlerstruktur vorgesehene Unterstützung wenden, nicht an die IBM Unterstützungsfunktion.

IBM