



Nutzung der erweiterbaren Informix-Technologie in einer serviceorientierten Architektur

Inhalt

- 2 *Kurzübersicht***
- 2 *Warum SOA?***
- 3 *Eine SOA in Aktion – hinter den Kulissen***
- 4 *Lose Kopplung – Schlüsselprinzip der SOA***
- 5 *Informix Dynamic Server und SOA***
- 6 *IDS als Provider von Web-Services***
- 7 *Erweiterbarkeit in IDS – ein Schlüsselfaktor für die SOA-Integration***
- 8 *IDS als Consumer von Web-Services***
- 9 *Nutzung von Web-Services in Informix-Anwendungen von ISVs***
- 10 *Fazit: Die Zeit ist reif, Informix Dynamic Server als Grundlage einer serviceorientierten Architektur zu nutzen***
- 10 *Anhang A: Empfohlene Produkte für die SOA-Integration von Informix***
- 11 *Anhang B: Zusatzmaterial***
- 11 *Weitere Informationen***

Kurzübersicht

Auf Grund des exponentiellen Wachstums ihrer IT-Systeme haben Unternehmen mit immer komplexeren Softwarearchitekturen zu kämpfen. Mehr denn je müssen IT-Abteilungen in kürzester Zeit auf neue Geschäftsanforderungen reagieren, die IT-Kosten für das Unternehmen reduzieren und neue Geschäftspartner und Kunden reibungslos in ihre Prozesse einbinden können. Währenddessen stoßen traditionelle Architekturen an die Grenzen ihrer Möglichkeiten. In der Softwarebranche bildeten sich im Laufe der Zeit verschiedene Architekturen für die Datenverarbeitung heraus, die eine vollständig verteilte Datenverarbeitung ermöglichen. Außerdem entstanden verschiedene Programmiersprachen, die für beliebige Plattformen geeignet sind und die Implementierungszeiten erheblich reduzieren sollten. Darüber hinaus wurden unzählige Verbindungsinfrastrukturen entwickelt, um eine bessere und schnellere Integration von Anwendungen zu erreichen.

Die Umstellung auf eine serviceorientierte Architektur (SOA) ist der nächste Schritt in dieser Evolution. Damit werden die IT-Organisationen dabei unterstützt, die immer komplexer werdenden Herausforderungen zu meistern. In diesem White Paper erfahren Sie, wie Sie eine SOA implementieren können, die Ihren individuellen Geschäftsanforderungen auf Basis von IBM Informix Dynamic Server (IDS) gerecht wird. IBM IDS ist ein strategischer IBM Datenserver, bei dem besonderes Gewicht auf leistungsfähiges OLTP (Online Transaction Processing), einfache Anwendungsintegration, geringen Administrationsbedarf und sehr niedrige Gesamtbetriebskosten (TCO) gelegt wird. IDS verfügt über eine moderne Prozessarchitektur mit hervorragender Erweiterbarkeit, die entscheidend ist für eine problemlose Integration in ein SOA-Framework.

Warum SOA?

Unternehmen können es sich heutzutage nicht leisten, in starren IT-Infrastrukturen gefangen zu sein. Wenn Mitarbeiter, Partner und Kunden auf interne Informationsquellen und -anwendungen zugreifen können, werden Unternehmen reaktionsfähiger und können besser von neuen Chancen auf dem Markt profitieren sowie Bedrohungen vermeiden.

Durch den Einsatz einer flexiblen Infrastruktur auf Basis einer serviceorientierten Architektur (SOA) sinken die Kosten für die Geschäftsabwicklung. Unternehmen können entsprechend den jeweiligen Geschäftsbedingungen praktisch im Handumdrehen neue Geschäftsbeziehungen eingehen und bestehende modifizieren. Wenn das Unternehmen seinem Geschäftsmodell entwächst, kann es sich im Grunde selbst neu erfinden – durch die Neukoordination lose gekoppelter Geschäftsprozesse und den Aufbau von Beziehungen mit neuen Partnern über einen veränderten Tooleinsatz und die Wiederverwendung bestehender Komponenten. All dies wird möglich, weil sich ihre Anwendungen und Geschäftsprozesse unter Verwendung lose gekoppelter Informationsservices, die auch Web-Services genannt werden, ändern und neu zusammenstellen lassen.

Mit Hilfe einer SOA können Unternehmen flexibel agieren, während die Kosten der Anwendungsimplementierung und -integration sinken. Diese Architektur ermöglicht die dynamische Interaktion von Web-Services. Da die SOA dazu beiträgt, die komplexen Abläufe in heterogenen IT-Umgebungen zu kaschieren, lassen sich bestehende Umgebungen – ob herkömmlicher oder moderner Art – kombinieren, um neue, komplexere Umgebungen und leistungsfähige, dynamische Anwendungen zu erstellen. SOA-basierte Anwendungen können in kürzester Zeit auf die dynamischen Einflüsse externer Umgebungen, von Märkten und der Wirtschaft reagieren. Durch eine SOA ist es für Unternehmen weit weniger teuer, ihre Geschäftsprozesse mit denen ihrer Lieferanten zu verknüpfen, um beispielsweise leistungsstarke Lieferketten aufzubauen. Kurz gesagt: Eine SOA unterstützt ein Unternehmen dabei, zu einem On Demand Business zu werden.

Eine SOA in Aktion – hinter den Kulissen

Nehmen wir einen Internetbuchhändler als Beispiel dafür, wie eine SOA funktioniert. Sie surfen durch den Internetbuchladen und kaufen vielleicht ein Buch. Dazu geben Sie Ihre Kreditkartendaten an, und ein paar Tage später erhalten Sie das bestellte Buch. Aus Ihrer Sicht eine einfache Transaktion, doch hinter den Kulissen war daran eine Reihe separater Services beteiligt, die zum Großteil nicht Eigentum des Online-Buchhändlers sind. So wurde vielleicht Ihre Adresse über den Service eines Drittanbieters geprüft, dessen Geschäft darin besteht, eine Vielzahl von Adressen zu kompilieren und die Adressen von Versandkunden zu validieren. Auch Ihre Kredit- oder EC-Kartennummer musste geprüft werden – ein weiterer Fremdservice. Die Bestellung ging an das Auslieferungslager, wurde ausgeführt und an ein Versandunternehmen wie UPS, FedEx oder DHL übergeben, das Ihr Paket zustellt.

Nur einige wenige der vielen Schritte, die zur Erledigung Ihrer Bestellung nötig waren, wurden von dem Onlinebuchhändler ausgeführt. Der Rest bestand aus einzelnen Services, die hinter den Kulissen über eine Schnittstelle verbunden werden, etwa über eine Anwendungsprogrammierschnittstelle (API). Die Kommunikation zwischen den einzelnen Prozessen erfolgte wahrscheinlich über den Austausch von Nachrichten im XML-Format. Sämtliche Einzelprozesse werden in Web-Services beschrieben. Die Koordination dieser Web-Services erfolgt über Software, die auf einem Anwendungsserver läuft.

Lose Kopplung – Schlüsselprinzip der SOA

Mit Hilfe einer SOA können komplexe Prozesse in kürzester Zeit die Arbeit herkömmlicher Anwendungen ausführen. Der Unterschied besteht darin, dass bei einer SOA die Komponentenservices lose gekoppelt und nicht permanent fest „verschweißt“ sind. Dieser Unterschied ist entscheidend, da das Unternehmen dadurch flexibel am Markt agieren kann.

Nehmen wir an, dass in unserem Beispiel des Internetbuchhändlers der Wettbewerbsdruck gestiegen oder die Nachfrage nach Büchern gesunken ist. Um diesen geänderten Marktbedingungen Rechnung zu tragen, erweitert der Buchhändler sein Onlineangebot um Streaming-Videos und Musik. Auf Grund der inhärenten Flexibilität der SOA lassen sich schnell neue Geschäftsprozesse zusammenstellen – bestehend aus wiederverwendbaren Komponenten, die auf unterschiedliche Weise koordiniert werden. So ist der Buchhändler in der Lage, mit neuen Geschäftspartnern zu kooperieren und deren Geschäftsprozesse zu integrieren. Der Buchladen kann sich also in einem Bruchteil der Zeit neu erfinden, die ohne eine SOA erforderlich gewesen wäre.

Ein Web-Service lässt sich als ein Remote-Funktionsaufruf beschreiben, der über Internet- oder Intranet-Kommunikationsprotokolle rund um XML-formatierte Nachrichten erfolgt. So wird der gegenseitige Zugriff über Firewalls hinweg möglich. Web-Services sind ein sehr wichtiger, aber nicht der einzig wichtige Bestandteil der SOA-Umgebung. XML-basierte Web-Services werden in der Regel auf Basis von SOAP (Simple Object Access Protocol) erstellt. Sie beschreiben sich zudem über die Web Service Description Language (WSDL). Es gibt darüber hinaus Verzeichnisse – Universal Description, Discovery and Integration (UDDI) genannt –, die sich wie „Gelbe Seiten“ verwenden lassen, um einen bestimmten Web-Service zu finden.

Die SOA und Web-Services sind heute zweifellos „in“. Ebenso wie die offenen Standards gibt es sie aber schon eine ganze Weile, so dass sich die Kombination einer SOA- und Web-Services-Umgebung mit älteren Anwendungen sehr einfach gestaltet.

Weshalb sollten Sie also jetzt eine SOA implementieren? Die Antwort auf diese Frage hat mit Risiken zu tun. Da die Technologie auf offenen Standards basiert, ist das Risiko, dass die SOA und die Web-Services obsolet werden, nur sehr gering. Und da die Technologie schon etabliert ist und die Entwickler die Anfangsprobleme bereits ausmerzen konnten, besteht nur ein geringes Risiko, dass es zu einem Misserfolg kommt. Unterdessen wurden bereits Best Practices entwickelt, mit deren Hilfe Sie Web-Services und die SOA noch effektiver nutzen können. Und schließlich wird auf dem Markt bereits eine Vielzahl von Softwareprodukten angeboten, die den Einstieg in eine SOA weiter vereinfacht. Eine der zentralen Grundlagen für Datenserver, die SOA unterstützen, ist IBM Informix Dynamic Server.

Informix Dynamic Server und SOA

Informix hat auf Grund seiner Stabilität, der hohen Leistung bei OLTP-Anwendungen und der niedrigen Betriebskosten bereits eine treue Anhängerschaft. Der besondere Vorteil der Verwendung von IDS in einer SOA liegt im Potenzial für „leichtgewichtige“, aber gleichzeitig leistungsstarke Integrationen, wie im Beispiel des Internethändlers. Durch die Kombination mit Systemen wie Anwendungen für die Kreditkartenprüfung werden sehr niedrige Einstiegskosten für ein Unternehmen möglich, das in diesem Bereich gerade erst am Anfang steht. Ein Einzelhandelsunternehmen mit interaktiven Anwendungen für die Auftragseingabe kann die Verwaltungskosten reduzieren und den Lieferzyklus verkürzen, indem es sein Auftragsverwaltungssystem für den Hersteller öffnet, der die Waren liefert. Durch die Integration von Informix-Anwendungen in einer SOA-Umgebung können Einzelhändler Bankssysteme miteinander verknüpfen, sich von der Punkt-zu-Punkt-Integration lösen und Ressourcen so rationell wie möglich nutzen. Es hat sich gezeigt, dass IDS eine sehr leistungsstarke Grundlage für den Aufbau einer SOA ist.

Damit eine Softwarekomponente eine erfolgreiche Grundlage für Web-Services bilden kann, muss sie als Anbieter (Provider) oder Konsument (Consumer) von Web-Services fungieren können. IDS kann beides sein. Für jede dieser Rollen ist eine eigene Methodik zur Ausführung des Vorgangs erforderlich.

IDS als Provider von Web-Services

Nehmen wir als Beispiel einen kleinen Produktionsbetrieb, der in das Auftragseingabesystem eines seiner größeren Kunden, einer weltweiten Einzelhandelskette, aufgenommen werden will. Der Produktionsbetrieb verwendet eine Anwendung auf Basis von IDS Version 7, und er würde es gerne ermöglichen, dass der Kunde Produkte direkt über sein Auftragseingabesystem bestellen kann.

Die Lösung besteht in diesem Fall für den Produktionsbetrieb in einem Upgrade auf IDS Version 10 und dessen Einsatz als Provider von Web-Services. Die Schnittstelle für diese Anwendung könnte ein Stored Procedure-Aufruf in IDS durch den Einzelhändler sein. Der Produktionsbetrieb verwendet das IBM Web Services Object Runtime Framework (WORF), um den Web-Service-Zugriff auf Stored Procedures von IDS zu ermöglichen. WORF wurde für IDS Version 10 zertifiziert, erweitert und getestet.

Über WORF wird eine einfache Web-Service-API zu einer Stored Procedure von IDS, einer SELECT-, INSERT- oder UPDATE-Anweisung oder allem bereitgestellt, das Sie nach außen hin als Web-Service anbieten möchten, der auf einfache Weise in eine Umgebung integrierbar ist. Eine Web-Service-Schnittstelle zu IDS lässt sich innerhalb von fünf bis zehn Minuten erstellen, je nachdem, wie vertraut Sie mit WORF sind. Für WORF ist ein J2EE™-kompatibler Anwendungsserver wie IBM WebSphere Application Server erforderlich.

WORF ist nicht die einzige Methode für die Bereitstellung von Web-Services. Dies ist auch über IBM Enterprise Generation Language (EGL) möglich. EGL weist sämtliche Markenzeichen einer strategischen Sprache für IBM auf. Sie basiert auf der Java-Technologie und verfügt über alle Stärken von Informix 4GL wie kurze Entwicklungszeiten, mächtige, aber dennoch einfach zu nutzende Sprachkonzepte und eine sehr gute Unterstützung der SQL-Datenbankabfragesprache. Darüber hinaus bietet Ihnen EGL leistungsstarke Technologien zur Entwicklung von Web-Services.

Neben WORF und EGL lässt sich Informix Dynamic Server auch einfach mit der Microsoft®.Net-Entwicklungsumgebung, Java, PHP, Python, Perl und vielen weiteren Technologien zur Bereitstellung von Web-Services kombinieren.

Erweiterbarkeit in IDS – ein Schlüsselfaktor für die SOA-Integration

IBM IDS umfasst zahlreiche Funktionen für die Erweiterung des Datenbankservers, einschließlich Unterstützung für neue Datentypen, Routinen, Aggregate und Zugriffsmethoden. Mit dieser Technologie lassen sich nicht nur auf Zeichen und numerischen Werten basierende Informationen erkennen und speichern, sondern über die entsprechenden Zugriffs- und Bearbeitungsrountinen auch nicht konventionelle Datenstrukturen verwalten, die entweder eher wie die Geschäftsumgebung gestaltet sind oder neue Datentypen enthalten, die zuvor noch nie für die Verarbeitung von Geschäftsanwendungen verfügbar waren. Auch wenn die Daten möglicherweise vom Standard abweichen und manche davon selbst tabellenartig sein können, werden sie auf relationale Weise mit Hilfe von Tabellen, Spalten und Zeilen gespeichert. Zudem erlauben alle Daten, Datenstrukturen, die über DDL-Befehle (Data Definition Language) erstellt wurden, sowie Zugriffsroutinen objektorientiertes Verhalten wie Überladung von Namen, Vererbung und Polymorphie. Diese objektorientierte Erweiterbarkeit unterstützt die Transaktionskonsistenz und Datenintegrität, und sie vereinfacht gleichzeitig die Datenbankoptimierung und -administration.

Andere Datenbankmanagementsysteme (DBMS) nutzen Middleware zur Verknüpfung mehrerer Server, von denen jeder unterschiedliche Datentypen verarbeitet, so dass das Ganze wie eine einzige Verarbeitungsumgebung erscheint. Durch dieses Konzept werden nicht nur die Leistung, sondern auch die Transaktionskonsistenz und -integrität beeinträchtigt, da Netzwerkprobleme zur Beschädigung der Daten führen können. Dies ist bei IBM IDS nicht der Fall. Seine objektrelationale Technologie ist in den Kern der Informix Dynamic Scalable Architecture eingebettet und kann je nach Bedarf im Rahmen einer Datenbankinstanz verwendet werden.

DataBlades. IBM Informix DataBlade-Module erweitern den Datenbankserver durch angepasste, benutzerdefinierte Datentypen, Routinen und Zugriffsmethoden mit zusätzlicher Geschäftsfunktionalität. Entwickler können diese neuen Datentypen und Routinen verwenden, um auf einfachere Weise Anwendungen mit umfassenden Funktionen zu erstellen und zu implementieren, die die Geschäftsanforderungen eines Unternehmens besser erfüllen. IBM IDS unterstützt die DataBlade-Funktionalität in gleichem Maße für integrierte oder benutzerdefinierte Routinen (UDRs) und Typen.

Durch IBM Informix DataBlade-Module lassen sich Informationen fast jeder Art einfach als Datentyp innerhalb des Datenbankservers verwalten. Informix unterstützt ein Reihe von DataBlade-Modulen anderer Hersteller. Die Entwickler können auch das IBM Informix DataBlade Developer's Kit (DBDK) verwenden, um angepasste Blades für eine bestimmte Geschäftsanforderung zu erstellen.

IDS als Consumer von Web-Services

Das oben genannte Beispiel zeigt, wie ein Unternehmen durch die Bereitstellung eines Web-Service Zugriff auf seine Informationsressourcen gewähren kann. Wenden wir uns nun einer anderen Situation zu, in der ein Hotel Zugriff auf eine Anwendung zur Kreditkartenvalidierung erlangen möchte. In dieser Situation ist das Hotel ein Consumer von Web-Services.

Dieses Hotel verfügt über eine herkömmliche Anwendung für die Zahlungsautorisierung und -annahme, die in Cobol erstellt wurde und in das Reservierungsmanagementsystem integriert ist. Die Anwendung muss um einen Web-Service für die Zahlungsautorisierung erweitert werden.

Die Lösung für die Nutzung von Web-Services wäre der Einsatz der erstklassigen Erweiterbarkeitsfunktionen von IDS entweder durch die Verwendung von IDS J/Foundation, einer leistungsstarken Umgebung, die die Integration von Java Stored Procedures in IDS ermöglicht, oder der DataBlade-API für C- und C++-basierende Routinen und benutzerdefinierte Datentypen. J/Foundation lässt sich sehr einfach kombinieren, z. B. mit dem Apache Axis Open Source SOAP-Framework. So können Entwickler eine IDS Java Stored Procedure (z. B. „Kreditkarteninhaber prüfen“) mit Programmlogik erstellen, der für den Zugriff auf einen Remote-Web-Service sorgt, um die Verbindung mit dem Unternehmen für die Kreditkartenzahlung herzustellen. Dies ist ein sehr gutes Beispiel von SQL-basierter Erweiterbarkeit, die Informix IDS so hervorragend unterstützt. Sie müssen keine größeren Änderungen an der Originalanwendung vornehmen, um die neue SQL-Routine zu verwenden. Sobald Sie die Java Stored Procedure ausführen, greift sie über das Internet auf den Remote-Web-Service zu und liefert die entsprechenden Ergebnisse zurück. Alternativ können C/C++-Entwickler stattdessen auch die IDS DataBlade-API verwenden und sie etwa mit dem gSOAP-Framework kombinieren, um dieselben Ergebnisse zu erzielen. Die erweiterbare Architektur von IDS bildet die perfekte Grundlage für den Zugriff auf Web-Services. Dabei werden klassische Informix-Anwendungen mit einem Web-Service zusammengeführt.

Nutzung von Web-Services in Informix-Anwendungen von ISVs

Tausende unabhängige Softwareanbieter (ISVs) nutzen seit Jahren Informix Dynamic Server und Informix 4GL zur Entwicklung von Anwendungen für ihre Kunden. Dies ist ein Markt, der enorm von den kürzeren Entwicklungszyklen profitieren kann, die durch den Einsatz von Web-Services möglich werden. IBM hat bereits zahlreiche Anfragen hinsichtlich der Web-Service-Fähigkeit bestehender Informix 4GL-Anwendungen erhalten.

ISVs sollten die Umstellung ihrer 4GL-Anwendungen auf EGL in Betracht ziehen und die leistungsfähige Web-Service-API von EGL nutzen. Im Rahmen von EGL können konvertierte 4GL-Funktionen (nach der Umstellung auf EGL) mit Unterstützung eines Anwendungsservers auf einfache Weise als Web-Services angeboten werden. Die Konvertierung von 4GL in EGL erfolgt im Grunde einfach über ein Konvertierungstool, das im EGL-Paket enthalten ist. Dadurch kann wertvolle Geschäftslogik aus der ursprünglichen 4GL-Anwendung weiterverwendet und mit EGL ein offener Web-Service erstellt werden, auf den andere zugreifen können.

Es gibt eine Reihe von Gründen für die Konvertierung Ihrer 4GL-Anwendungen in EGL-Anwendungen:

- *EGL ermöglicht die einfache Bereitstellung von Web-Services und den Zugriff darauf. Die Unterstützung für Web-Services ist integraler Bestandteil der EGL-Sprache.*
- *EGL ist eine einfach zu erlernende, 4GL-ähnliche Sprache auf Basis der Java-Technologie.*
- *EGL unterstützt zeichenbasierte (4GL-ähnliche), Batch-, Web-Anwendungen und Eclipse Rich Client Platform (RCP) basierte GUI-Anwendungen.*
- *Ein im Lieferumfang enthaltenes Konvertierungsdienstprogramm sorgt für eine einfache Konvertierung von 4GL in EGL.*
- *EGL unterstützt Microsoft Windows[®] sowie Linux[®] und UNIX[®].*

Fazit: Die Zeit ist reif, Informix Dynamic Server als Grundlage einer serviceorientierten Architektur zu nutzen

Serviceorientierte Architekturen und Web-Services sorgen für die einfache Integration heterogener Anwendungen. Im Rahmen des IBM Softwareportfolios für das Informationsmanagement werden Datenservices bereitgestellt, die allen aktuellen und zukünftigen Kundenanforderungen im Hinblick auf die SOA-Entwicklung gerecht werden. Anwendungen, die auf IBM Informix basieren, lassen sich problemlos in bestehende SOA-Umgebungen integrieren. Dank ihrer erstklassigen erweiterbaren Architektur schafft Version 10 von IDS die notwendige Grundlage für eine SOA. Kunden, die jetzt handeln, können herkömmliche Informix-Anwendungen (z. B. auf Basis von 4GL) einfach in Services umwandeln und damit ihre Investitionen schützen.

Weitere Informationen

Weitere Informationen erhalten Sie von Ihrem IBM Vertriebsbeauftragten oder IBM Business Partner.

Anhang A:

Empfohlene Produkte für die SOA-Integration von Informix

- *IBM IDS, Version 10 (jede Edition).*
 - *Enthält Erweiterungen für eine bessere SOA-Integration.*
 - *Bietet die nötige erstklassige Erweiterbarkeitstechnologie, durch die IDS über beliebige benutzerdefinierte Routinen, die in Java und/oder C/C++ erstellt wurden, Web-Services anbieten oder nutzen kann.*
- *IDS, Version 10 Erweiterungen (DataBlades).*
 - *WebSphere MQ DataBlade: Verknüpfung von IDS mit einer IBM WebSphere MQ-Messaging-Umgebung, die als Grundlage für Web-Service-Nachrichten dienen kann.*
 - *Das Web DataBlade ermöglicht die Erstellung anwendungsspezifischer XML-Dokumente. Nachrichten und Dokumente im XML-Format sind die Lingua franca der Web-Services.*
 - *Das „XML Generating UDRs DataBlade“ (Informix DeveloperZone).*
 - *Das „XSLT DataBlade“ (Alphaworks): einfache XML-Dokumentenumwandlung.*

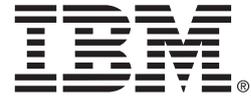
- *WORF (Web Services Object Runtime Framework): einfacher Web-Service-Zugriff auf IDS SQL-Anweisungen wie SELECT, UPDATE, INSERT, DELETE und Aufrufe von Stored Procedures.*
 - *Teil der IBM Rational Software Development Platform (SDP).*
 - *Teil von WebSphere Application Server.*
- *IBM Enterprise Generation Language (EGL).*
 - *Teil der Rational SDP (z. B. Rational Application Developer).*
 - *Integriertes Tool für die Konvertierung von 4GL in EGL.*
 - *Integrierte Web-Service-API für die Bereitstellung und Nutzung von Web-Services auf Anwendungsebene.*
- *Rational Development Tools/WebSphere Application Server (XML Input, XML Output): generisches Toolset auf Basis der Eclipse- und Java-Technologie sowie Java-Anwendungsserver mit vielen Tools und Assistenten zur Entwicklung und Implementierung von Web-Services jeder Art auf Grundlage von IDS.*
- *IBM Informix .NET Provider.*
 - *Einige Entwickler bevorzugen vielleicht die Microsoft .NET-Umgebung und deren Web-Service-Unterstützung für diesen Zweck. Diese Entwickler können den IBM Informix .NET Provider verwenden, um IDS mit .NET-basierten Web-Services zu integrieren.*
- *JAXP-Unterstützung des Informix JDBC 3.0-Treibers (XML-Validierung während des Lesens oder des Updates eines in IDS gespeicherten XML-Dokuments).*
- *Excalibur Text Search DataBlade: Volltextsuche auch für XML-Dokumente.*
- *Spatial DataBlade 8.20: XML-formatierte Ausgabe von raumbezogenen Geometriedaten – Geography Markup Language (GML).*

Anhang B:

Zusatzmaterial

- *IBM Redbook Powering SOA with IBM Data Servers, SG24-7259-00, Link: www.redbooks.ibm.com/abstracts/sg247259.html?Open*
- *IBM Redbook Informix Dynamic Server V10 – Extended Functionality for Modern Business, SG24-7299-00, Link: www.redbooks.ibm.com/abstracts/sg247299.html?Open*

Weitere Informationen zu IBM Redbooks erhalten Sie online unter: www.redbooks.ibm.com



IBM Deutschland GmbH
70548 Stuttgart
ibm.com/de

IBM Österreich
Obere Donaustraße 95
1020 Wien
ibm.com/at

IBM Schweiz
Vulkanstrasse 106
8010 Zürich
ibm.com/ch

Die IBM Homepage finden Sie unter:

ibm.com

IBM, das IBM Logo und ibm.com sind eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern.

DataBlade, Informix, Rational und WebSphere sind Marken der IBM Corporation in den USA und/oder anderen Ländern.

Java und alle Java-basierenden Marken und Logos sind Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

Linux ist eine Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und/oder anderen Ländern.

Hinweise auf IBM Produkte, Programme und Services in dieser Veröffentlichung bedeuten nicht, dass IBM diese in allen Ländern, in denen IBM vertreten ist, anbietet.

Hergestellt in den USA
10-06

© Copyright IBM Corporation 2006
Alle Rechte vorbehalten.

TAKE BACK CONTROL WITH **Information Management**