



ObjectBridge EJB Edition für BEA Weblogic Tuxedo Connector

Enterprise JavaBeans (EJB) – ein Begriff, der im Bereich unternehmenskritischer Anwendungen mehr und mehr auf Resonanz stößt. Als serverseitige Komponentenarchitektur für den unternehmensweiten Einsatz unterstützt Sie EJB bei der zukunftsicheren Realisierung Ihrer Geschäftsprozesse. ObjectBridge EJB Edition beschleunigt und erleichtert die Entwicklung dieser erfolgsentscheidenden Anwendungen, sowohl bei der Einbindung bestehender Anwendungen und als auch für den Einsatz im Internet.

ObjectBridge EJB Edition für BEA Weblogic Tuxedo Connector (WTC) wird als Zusatzkomponente unter SCORE/Integration Suite eingesetzt und ermöglicht den objektbasierten Zugriff von Java- oder HTML-Clients auf alle Server-Komponenten, die unter der Middleware Tuxedo lauffähig sind. Dabei stellt BEA Weblogic Tuxedo Connector (WTC) die Verbindung zwischen dem BEA Weblogic Server und den Tuxedo Services über das Java Application to Transaction Monitor Interface (JATMI) her. Die von der ObjectBridge EJB Edition generierten EJB-Proxies bedienen sich dieser JATMI-Schnittstelle um mit den zugehörigen Server-Komponenten zu korrespondieren, die als Tuxedo Services implementiert sind.

INHALT

Proxy-Generierung	3
Vom Client zum Server.....	3
Performance-Steigerung.....	4
Die Komponenten der ObjectBridge	4
Der ObjectBridge Generator	5
Das Archiv der Java-Basisklassen	5
Verfahren.....	5
Generierung.....	6
Kompilierung.....	6
Deployment.....	6
Generierte Klassentypen.....	7
Die Server-Klasse	7
Die Interface-Klasse	7
Die Operation-Parameter-Klasse	7
Systemvoraussetzungen	8

PROXY-GENERIERUNG

ObjectBridge EJB Edition für WTC generiert EJB-Proxies, zusammen mit den für das Deployment benötigten Deployment Descriptoren, als vollständige und lauffähige Enterprise JavaBeans.

Mit Hilfe dieser generierten EJB-Proxies greift eine Client-Anwendung auf eine beliebige Server-Komponente zu, die als Tuxedo-Service installiert ist. Dabei müssen die Java-Programmierer nicht mehr wissen, wie eine Komponente tatsächlich implementiert wurde, da die Server über generierte Java-Klassen (EJB-Proxies) aufgerufen werden.

Indem die EJB-Proxies die Methodenaufrufe automatisch in entsprechende Server-Requests umsetzen, wird eine Serverkomponente so angesprochen, als wäre sie selbst eine in Java geschriebene Klasse mit zugehörigen Methoden.

Dabei setzen die generierten Klassen die Java-Datentypen automatisch in die entsprechenden Datentypen der Server-Seite bzw. umgekehrt um.

VOM CLIENT ZUM SERVER

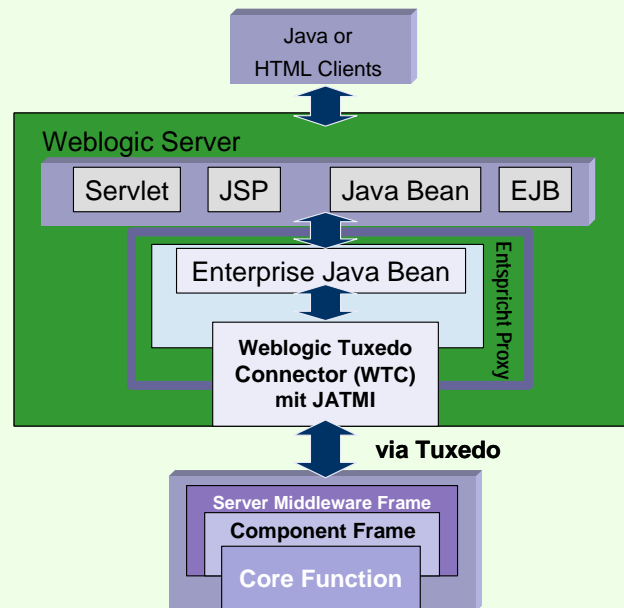
Der Zugriff auf eine Server-Komponente wird über Enterprise JavaBeans (Session Beans) realisiert, die mittels ObjectBridge EJB Edition automatisch generiert werden. Direkt nachdem die EJBs kompiliert und deployed wurden, können sie im BEA Weblogic Server verwendet werden.

Über die Methoden der in einer Client-Anwendung verwendeten Enterprise JavaBeans kann anschließend die Server-Komponente aufgerufen werden, ohne dass die Client-Anwendung die Implementierung der Server-Komponente kennen muss. Dabei werden lediglich die äußeren Schnittstellen (Interfaces) der Server-Komponente verwendet, die die generierten EJBs als Methoden zur Verfügung stellen.

Mit Hilfe des BEA Weblogic Tuxedo Connectors (WTC) wird über das Java Application to Transaction Monitor Interface (JATMI) eine Verbindung zur Middleware Tuxedo aufgebaut, um von dort über den Server Middleware Frame (SMF) und den Server Component Frame (SCF) zum eigentlichen Server-Modul zu gelangen. Die aufrufenden EJB-Proxies, die die entsprechenden JATMI-Aufrufe enthalten, ebenso wie der Middleware und der Component Frame für die Server-Komponente werden von SCORE/Integration Suite zur Verfügung gestellt bzw. von dort aus generiert.

Die strikte Schichtenarchitektur und die Generatoren erleichtern und beschleunigen die Entwicklung von Anwendungen beträchtlich. Außerdem sichern sie für Client- und Server-Komponenten die Unabhängigkeit von technischen Änderungen, z.B. bei Verwendung von Request/Response Packaging zur Steigerung der Performance und schützen so Ihre Investitionen.

Folgende Grafik verdeutlicht die Technik der Kommunikation zwischen Client und Server:



Struktur einer verteilten Anwendung mit ObjectBridge EJB Edition für BEA Weblogic Tuxedo Connector

PERFORMANCE-STEIGERUNG

Um die Anzahl der Aufrufe zwischen Client und Server zu reduzieren, können für eine Komponente Optimierungen in Form von Request- und Response-Packaging in der Definition der Server-Schnittstellen spezifiziert werden. Dabei führen die EJB-Proxies das Packaging automatisch durch und steigern somit die Performance der Anwendungen, ohne dass diese verändert werden müssen.

DIE KOMONENTEN DER OBJECTBRIDGE

Die ObjectBridge EJB Edition für WTC besteht aus:

- dem Generator für die Erzeugung der Enterprise JavaBeans und
- dem Archiv der Java-Basisklassen.

Des weiteren muss folgende Software zur Verfügung stehen:

- BEA Weblogic Server ab Version 6.1
- BEA Weblogic Tuxedo Connector ab Version 1.1

DER OBJECTBRIDGE GENERATOR

Der ObjectBridge Generator ist verantwortlich für die automatische Erstellung der Enterprise JavaBeans. Ausgangspunkt für die Generierung dieser Klassen ist ein Component Repository, das die Definitionen der Serverkomponente, d.h. die Beschreibung der Schnittstellen, Objekte, Methoden, Compound Operations, etc. enthält. Die Informationen des Component Repository sind im XML-Format gespeichert und können deshalb auch mit anderen Werkzeugen ausgewertet werden.

Der ObjectBridge Generator interpretiert diese Informationen und erzeugt daraus Java-Klassen (Proxies) als Enterprise JavaBeans.

DAS ARCHIV DER JAVA-BASISKLASSEN

ObjectBridge EJB Edition enthält bereits vorgefertigte Java-Basisklassen. Diese werden von den generierten Java-Proxies verwendet und enthalten u.a. Methoden zur Konvertierung von Datentypen sowie Exception-Klassen für die Fehlerfallbehandlung.

Der BEA Weblogic Tuxedo Connector (WTC) bietet mit dem Java Application to Transaction Monitor Interface (JATMI) die Möglichkeit, eine Verbindung zwischen dem BEA Weblogic Server und Tuxedo Services herzustellen.

Die mit ObjectBridge EJB Edition für WTC generierten Enterprise JavaBeans enthalten die entsprechenden JATMI-Aufrufe um über Tuxedo mit der Server-Komponente zu kommunizieren. Die Tuxedo-spezifischen JATMI-Aufrufe werden in den EJB-Proxies gekapselt. Somit kann eine Client-Anwendung entwickelt werden, ohne Middleware-spezifische Besonderheiten und Anforderungen berücksichtigen zu müssen.

VERFAHREN

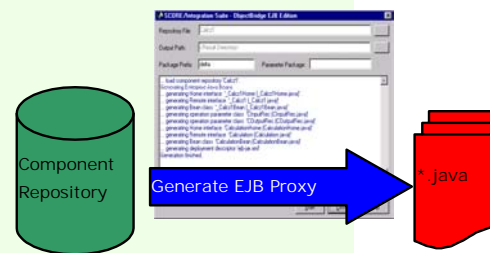
Vorgehensweise:

- Generierung der Java-Klassen
- Kompilierung der Enterprise JavaBeans und Archivbildung
- Deployment der gebildeten Packages

Bei Verwendung des Java Development Kit (JDK) von Sun oder darauf basierender Java-Entwicklungsumgebungen kann der komplette Entwicklungsprozess einschließlich der Kompilation der Java-Klassen innerhalb von Scout² durchgeführt werden.

GENERIERUNG

Basis für die Generierung der EJB-Proxies ist ein Component Repository, das alle relevanten Informationen zu einer Server-Komponente und ihren Schnittstellen enthält. Die Generierung erzeugt automatisch die entsprechenden Enterprise JavaBeans sowie einen dazu gehörigen Deployment Descriptor.



Erzeugung EJB-Proxies

KOMPILIERUNG

Die so entstandenen Enterprise JavaBeans werden nun kompiliert und nachfolgend mit dem Deployment Descriptor zu einem Package (Java-Archiv) zusammengefasst.

Erfolgt dabei die Erstellung des Package direkt unter der Development Middleware Scout², wird die genaue Abbildung der Package-Struktur auf die für die weitere Verwendung erforderliche Verzeichnisstruktur automatisch durchgeführt, ebenso wie die korrekte Ablage des Deployment Descriptors.

Wenn Sie das Java Development Kit (JDK) von Sun verwenden, können Sie die Kompilierung direkt in Scout² ausführen. Beim Einsatz anderer Werkzeuge (z.B. IBM Visual Age) müssen Sie die erzeugten Java-Sourcen in Ihre Entwicklungsumgebung importieren.

DEPLOYMENT

Für die Verwendung der EJBs durch Java-Clients werden die erstellten Packages in den BEA Weblogic Server deployed.

Das herausragende Merkmal dieses Verfahrens ist, dass die Client-Anwendungen mit Hilfe der generierten, kompilierten und deployten EJB-Proxies die Funktionen des entsprechenden Servers aufrufen als wären diese selbst in Java geschrieben.

GENERIERTE KLASSENTYPEN

Der ObjectBridge Generator erzeugt aus dem Component Repository die folgenden Java-Klassen:

- Die Server-Klasse
- Die Interface-Klasse
- Die Operation-Parameter-Klasse

DIE SERVER-KLASSE

Für jede Server-Komponente wird eine Server-Klasse einschließlich eines Remote- und Home-Interfaces als Enterprise JavaBean generiert.

Die Server-Klasse verarbeitet alle eingehenden Java-Methodenaufrufe, bildet aus diesen Compound Operations und ruft die zugehörige Server-Komponente auf. Außerdem ist sie für die optimierte Ausführung der Operationen, das Request-/Response-Packaging, zuständig.

DIE INTERFACE-KLASSE

Ebenso wird für jedes Interface einer Server-Komponente eine Enterprise JavaBean, einschließlich eines zugehörigen Remote- und Home-Interfaces, erzeugt. Diese EJB enthält Java-Methoden für alle Operationen, die das Server-Interface zur Verfügung stellt.

DIE OPERATION-PARAMETER-KLASSE

Für jeden in einer Operation als Struktur definierten Parameter erzeugt der Generator ebenfalls eine eigene Klasse, die Operation-Parameter-Klasse. Eine solche Java-Klasse enthält alle Datenelemente des Operationsparameters als public-Felder vom Typ `String`. Die Felder werden in beiden Richtungen automatisch in die von der Gegenseite benötigten Datentypen konvertiert.

SYSTEMVORAUSSETZUNGEN

... FÜR DIE GENERIERUNGSUMGEBUNG:

Hardware IBM-kompatibler PC mit Prozessor der Pentium-Klasse
128 MB Hauptspeicher (minimal)

Betriebssysteme Microsoft Windows® NT 4.0 / 2000 / XP

Software SCORE/Integration Suite
Delta/Scout²

... FÜR DIE LAUFZEITUMGEBUNG:

Betriebssysteme Microsoft Windows® NT 4.0 / 2000 / XP
Sun Solaris®
IBM AIX
Weitere Unix Systeme auf Anfrage

Software Sun JDK 1.3 oder höher kompatible Java VM
BEA Weblogic Server 6.1 oder höher
BEA Weblogic Tuxedo Connector (WTC)

MA 13959.03

www.d-s-t-g.com