

ObjectBridge Java Edition

Als Bestandteil von SCORE Integration Suite stellt ObjectBridge Java Edition eine Verbindung von einem objektorientierten Java-Client zu einer fast beliebigen Server-Komponente her. ObjectBridge Java Edition generiert Java-Klassen (Proxies), über die eine in Java entwickelte Client-Anwendung einen objektorientierten (z.B. Java- oder C++-Server) oder auch nicht objektorientierten Server (z.B. COBOL-Programm) ansprechen kann.

INHALT

ObjectBridge Java Edition	3
Proxy Generierung	3
Middleware unabhängig	3
ObjectBridge-Komponenten	4
Der ObjectBridge-Generator	4
Das Archiv der Java Basisklassen.....	4
Middleware Interface.....	5
Verfahren.....	6
Generierung	6
Compile-Vorgang.....	6
Performance Steigerung	6
Generierte Klassentypen.....	7
Die Serverklasse.....	7
Die Interface Klasse	7
Die Operation Parameter Klasse	7
Systemvoraussetzungen	8
... für die Generierungsumgebung:	8
... für die Laufzeitumgebung:	8

OBJECTBRIDGE JAVA EDITION

ObjectBridge Java Edition als Zusatzkomponente zu SCORE Integration Suite bietet einen objektbasierten Zugriff auf alle Komponenten, die mit SCORE Integration Suite integriert wurden. Es spielt dabei keine Rolle, ob die Komponente selbst mit Hilfe objektorientierter Werkzeuge bzw. Programmiersprachen entwickelt wurde oder nicht.

PROXY GENERIERUNG

Auf der Client-Seite ist ein mit Hilfe der von ObjectBridge Java Edition generierten Java-Klassen entwickeltes Java-Anwendungsprogramm in der Lage, auf eine beliebige mit SCORE Integration Suite erstellte Server-Komponenten zuzugreifen.

Für den Java-Programmierer ist es nicht mehr ersichtlich, wie diese Komponente implementiert wurde, da der Aufruf der Server über die generierten Java-Klassen (Proxies) erfolgt und der Entwickler nur die äußeren Schnittstellen (Interfaces) der Komponente verwendet.

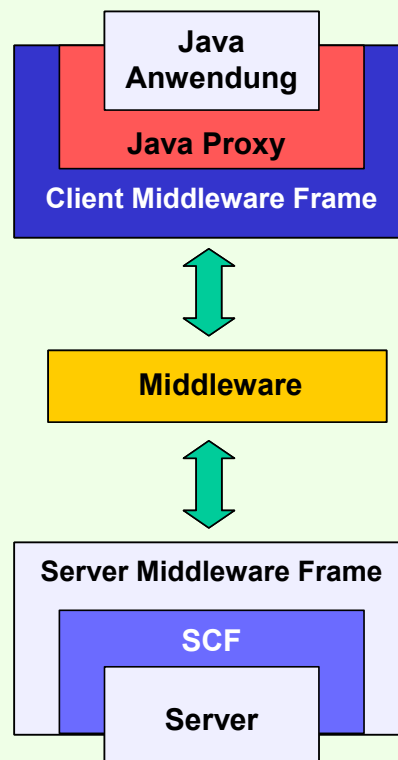
Die Proxies setzen ihre Methodenaufrufe in entsprechende Server-Requests um. Der Java-Programmierer kann die Server-Komponente so ansprechen, als wäre sie eine in Java geschriebene Klasse mit zugehörigen Methoden.

Die Konvertierung der Java-Datentypen in Datentypen auf der Server-Seite und umgekehrt erfolgt hierbei automatisch in den generierten Klassen.

MIDDLEWARE UNABHÄNGIG

Für das Anwendungsprogramm spielt es keine Rolle, ob der Server auf dem gleichen Rechner ausgeführt wird, oder ob er auf einem entfernten System läuft und über eine Middleware angesprochen wird. Die Client-Anwendung kann also zunächst komplett lokal entwickelt werden. Danach kann man die zugehörige Serverkomponente über das transparente SCORE Middleware Target aufrufen, ohne dass eine Änderung des Java-Clients notwendig wäre.

Aus technischer Sicht kommunizieren die generierten Java-Proxies über den Client Middleware Frame (CMF) mit der entsprechenden Middleware und diese über den Server Middleware Frame (SMF) und den Server Component Frame (SCF) mit dem eigentlichen Server-Modul.



Struktur einer Client/Server-Anwendung mit Java Proxies.

CMF, SMF und SCF werden von SCORE Integration Suite zur Verfügung gestellt bzw. generiert (siehe Abbildung rechts).

OBJECTBRIDGE-KOMPONENTEN

ObjectBridge Java Edition besteht aus folgenden Komponenten:

- **Dem Generator für die Erzeugung von Java-Klassen**
- **Das Archiv der Java-Basisklassen**
- **Middleware Interface für lokale Server Aufrufe**

DER OBJECTBRIDGE-GENERATOR

Ausgangspunkt für den ObjectBridge-Generator ist das Composition Model, das die Definitionen der Serverkomponente beinhaltet (u.a. Beschreibung der Schnittstellen, Objekte, Methoden, Signaturen etc.).

Der ObjectBridge-Generator interpretiert diese Informationen und erstellt die entsprechenden Java-Klassen (Proxies).

DAS ARCHIV DER JAVA BASISKLASSEN

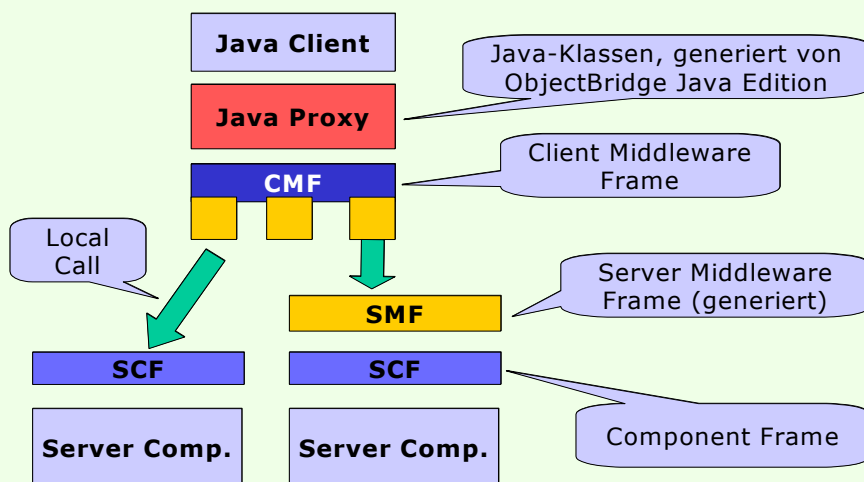
Zusammen mit ObjectBridge Java Edition werden vorgefertigte Java-Klassen als jar-Archiv ausgeliefert. Diese Klassen werden von den generierten Java-Proxy-Klassen verwendet und enthalten u.a. Datentyp-Konvertierungs-Methoden, Methoden zur Transaktionssteuerung und Exception-Klassen für den Fehlerfall.

MIDDLEWARE INTERFACE

Der Aufruf entfernter Server ist abhängig von der verwendeten Middleware. Für lokale Serveraufrufe und für eine Middleware, für die es eine Java-Schnittstelle (z.B. IBM CICS Java Gateway) gibt, wird die Verbindung komplett in Java zur Verfügung gestellt. Ansonsten werden zusammen mit dem SCORE Middleware Target zusätzliche Bibliotheken als Schnittstelle zu der entsprechenden Middleware mitgeliefert.

Eine Client-Anwendung kann Middleware-unabhängig entwickelt werden und bei einem Austausch der Middleware sind somit keine Änderungen des Java-Clients nötig.

Besondere Middleware-spezifische Einstellungen für einen Server bzw. einen Host können in einer Konfigurations-Datei hinterlegt und zur Laufzeit ausgewertet werden.

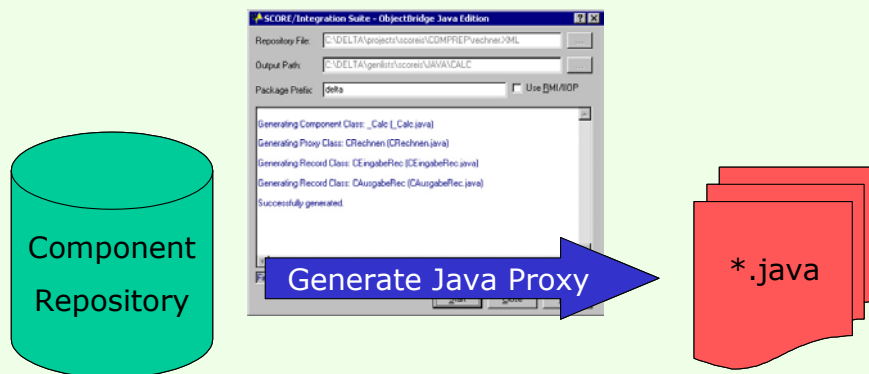


Struktur des Aufrufs eines lokalen und eines entfernten Servers von einer mit SCORE Integration Suite entwickelten Java-Anwendung.

VERFAHREN

GENERIERUNG

Ausgangspunkt für die Generierung ist das Composition Model (XML-Datei). Dieses Daten-File beschreibt eine Serverkomponente und deren Schnittstellen. Die daraus generierten Java-Klassen werden zu einem Package zusammengefasst. Diese Klassen müssen anschließend compiliert werden, um sie von der Client-Anwendung ansprechen zu können.



Erzeugung der Java-Klassen (Proxies) auf der Grundlage des Component Repository.

COMPILE-VORGANG

Wenn das JDK von Sun verwendet wird, kann der Compile-Vorgang direkt in Delta/Scout² ausgeführt werden. Bei anderen Werkzeugen (z.B. IBM Visual Age) werden die erzeugten Java-Sourcen ggf. in die Entwicklungsumgebung importiert.

Die Client-Anwendung kann anschließend mit Hilfe der generierten und compilierten Java-Proxies die Funktionen des Servers über die Methoden dieser Proxies aufrufen.

PERFORMANCE STEIGERUNG

Um die Anzahl der Aufrufe zwischen Client und Server zu reduzieren, können Optimierungen in Form von Request- und Response-Packaging im Repository für eine Komponente spezifiziert werden. Das Packaging wird von den Proxies unterstützt und kann somit die Performance der Anwendung steigern.

GENERIERTE KLASSENTYPEN

Der ObjectBridge-Generator erzeugt zu jedem Composition Model einer Server-Komponente folgende Java Klassen:

DIE SERVERKLASSE

Pro Serverkomponente wird eine Serverklasse generiert. Diese bündelt alle Java Methodenaufrufe, bildet daraus Signaturen und ruft die zugehörige Serverkomponente auf. Zusätzlich ist sie zuständig für das optimierte Ausführen (Request-/Response-Packaging) von Operationen.

DIE INTERFACE KLASSE

Pro Interface einer Komponente wird eine Java Klasse erzeugt. Diese Klasse enthält Java Methoden für alle Operationen, die das Interface zur Verfügung stellt.

DIE OPERATION PARAMETER KLASSE

Für jeden aus einer Struktur bestehenden Parameter einer Operation erzeugt der Generator eine eigene Klasse (Operation Parameter Klasse). Diese Java Klasse enthält alle Data Elements des Operation Parameters als public Felder vom Typ String. Die Felder werden automatisch in die Datentypen der Serverkomponente konvertiert und umgekehrt.

SYSTEMVORAUSSETZUNGEN

... FÜR DIE GENERIERUNGSUMGEBUNG:

Hardware	IBM-kompatibler PC mit Prozessor der Pentium-Klasse 64 MB Hauptspeicher (minimal)
Betriebssysteme	Microsoft Windows® NT / 2000 / XP
Software	SCORE Integration Suite ab 1.5 SCOUT²

... FÜR DIE LAUFZEITUMGEBUNG:

Betriebssysteme	Microsoft Windows® NT / 2000 / XP Sun Solaris® Weitere Unix Systeme auf Anfrage
Software	Sun JDK 1.1.6 oder höher kompatible Java VM

MA 13'956.03

www.d-s-t-g.com