SCORE
Integration Suite

DELTA
software
technology

ObjectBridge für C++

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

© 2003 Delta Software Technology GmbH

INHALT

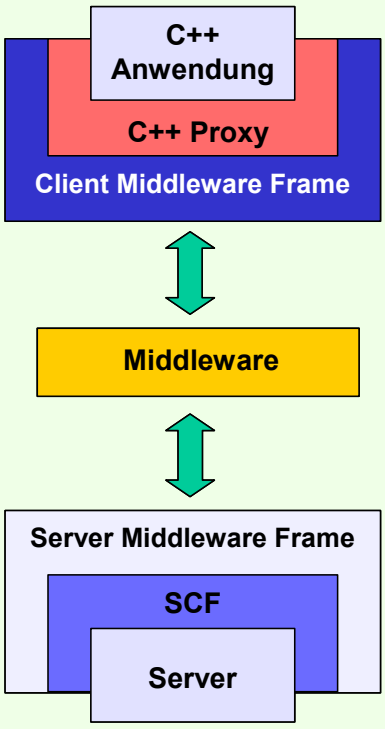
ObjectBridge für C++ 3
Proxy Generierung 3
Middleware unabhängig 3
ObjectBridge-Komponenten 4
Der ObjectBridge-Generator 4
Die Library mit den C++ Basisklassen 5
Runtime-Bibliothek 5
Middleware Interface 5
Verfahren 6
Generierung 6
Compile-Vorgang 6
Performance Steigerung 6
Generierte Klassentypen 7
Die Serverklasse 7
Die Interface-Klasse 7
Die OperationParameter-Klasse 8
Systemvoraussetzungen 8
...für die Generierungsumgebung 8
...für die Laufzeitumgebung 8

Delta Software Technology GmbH behält sich alle Rechte für Änderungen und Verbesserungen am Produkt vor.

Alle Rechte an dieser Dokumentation sind vorbehalten. Ohne schriftliche Genehmigung durch Delta Software Technology GmbH darf dieses Dokument weder als Ganzes noch in Teilen kopiert, photokopiert, reproduziert, übersetzt oder in irgend eine elektronische Form überführt werden.

Delta und SCORE sind eingetragene Warenzeichen der Delta Software Technology GmbH. Andere Warenzeichen sind registrierte Warenzeichen der jeweiligen Eigentümer.

OBJECTBRIDGE FÜR C++



Als Bestandteil von SCORE Integration Siute stellt ObjectBridge für C++ eine Verbindung von einem objektorientierten C++-Client zu einer fast beliebigen Server-Komponente her. ObjectBridge für C++ generiert C++-Klassen (Proxies), über die eine in C++ entwickelte Client-Anwendung einen objektorientierten (z.B. Java- oder C++-Server) oder auch einen nicht objektorientierten Server (z.B. COBOL-Programm) ansprechen kann.
ObjectBridge für C++ als Zusatzkomponente zu SCORE Integration Siute bietet einen objektbasierten Zugriff auf alle Komponenten, die mit SCORE Integration Siute integriert wurden. Es spielt dabei keine Rolle, ob die Komponente selbst mit Hilfe objektorientierter Werkzeuge bzw. Programmiersprachen entwickelt wurde oder nicht.

Struktur einer Client/Server-Anwendung mit C++ Proxies

PROXY GENERIERUNG

Auf der Client-Seite ist ein mit Hilfe der von ObjectBridge für C++ generierten C++-Klassen entwickeltes C++-Anwendungsprogramm in der Lage, auf eine beliebige mit SCORE Integration Siute erstellte Server-Komponente zuzugreifen. Für den C++-Programmierer ist es nicht mehr ersichtlich, wie diese Komponente implementiert wurde, da der Aufruf der Server über die generierten C++-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 C++-Programmierer kann die Server-Komponenten so ansprechen, als wären es in C++ geschriebene Klassen mit zugehörigen Methoden.

Die Konvertierung der C++-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 Targetaufrufen, ohne dass eine Änderung des C++-Clients notwendig wäre.

Aus technischer Sicht kommunizieren die generierten C++-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. CMF, SMF und SCF werden von SCORE Integration Siute zur Verfügung gestellt bzw. generiert (siehe Abbildung in Kapitel "ObjectBridge für C++").

OBJECTBRIDGE-KOMPONENTEN

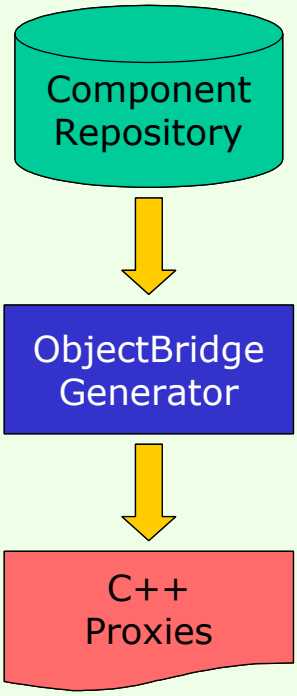
ObjectBridge für C++ besteht aus folgenden Komponenten:

- dem Generator für die Erzeugung der C++-Klassen
- der Library mit den C++-Basisklassen
- Runtime-Bibliothek
- Middleware Interface für lokale Serveraufrufe

DER OBJECTBRIDGE-GENERATOR

Ausgangspunkt für den ObjectBridge-Generator ist das Component Repository, 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 C++-Klassen (Proxies).



Ablauf der Generierung mit ObjectBridge für C++

DIE LIBRARY MIT DEN C++ BASISKLASSEN

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

RUNTIME-BIBLIOTHEK

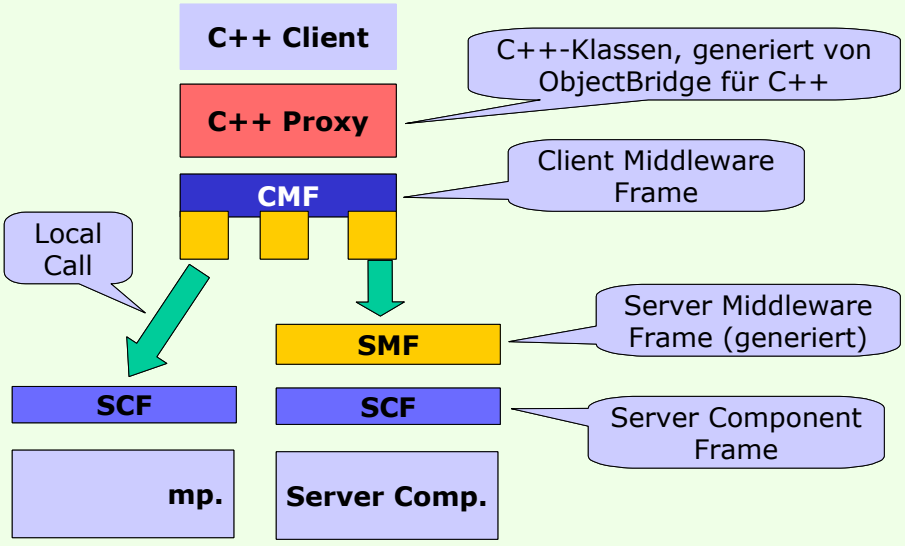
Die mitgelieferte Runtime-Bibliothek enthält Funktionen für die Konvertierung der Datentypen auf der Serverseite in C-Datentypen und umgekehrt.

MIDDLEWARE INTERFACE

Der Aufruf entfernter Server ist abhängig von der verwendeten Middleware. Zusammen mit dem SCORE Middleware Targetwerden 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 C++-Clients nötig.

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

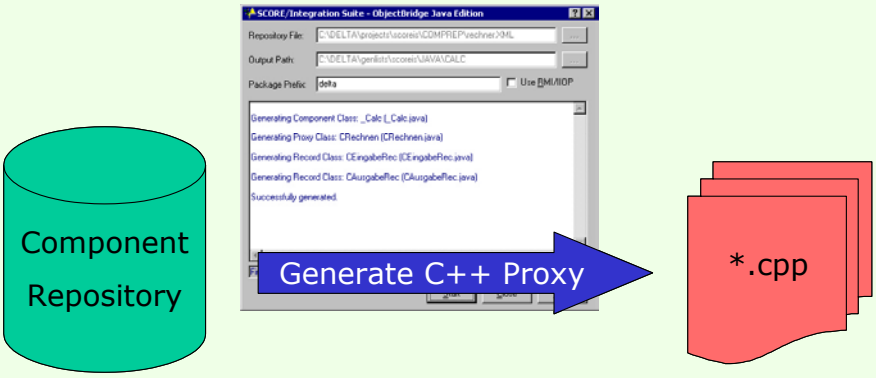


Struktur des Aufrufs eines lokalen und eines entfernten Servers von einer mit SCORE Integration Siute entwickelten C++ Anwendung

VERFAHREN

GENERIERUNG

Ausgangspunkt für die Generierung ist das Component Repository (XML-Datei). Dieses Repository beschreibt eine Serverkomponente und deren Schnittstellen. Die daraus generierten C++-Klassen müssen anschließend compiliert und mit der Client-Anwendung gelinkt werden, um sie ansprechen zu können.



Erzeugung der C++ Klassen (Proxies) auf der Grundlage des Component Repository

COMPILE-VORGANG

Wenn nur ein C++-Compiler installiert ist, kann der Compile-Vorgang direkt in Delta/Scout² ausgeführt werden. Bei anderen C++-Entwicklungswerkzeugen (z.B. Microsoft Visual C++) werden die erzeugten C++-Sourcen in die entspr. Umgebung importiert.

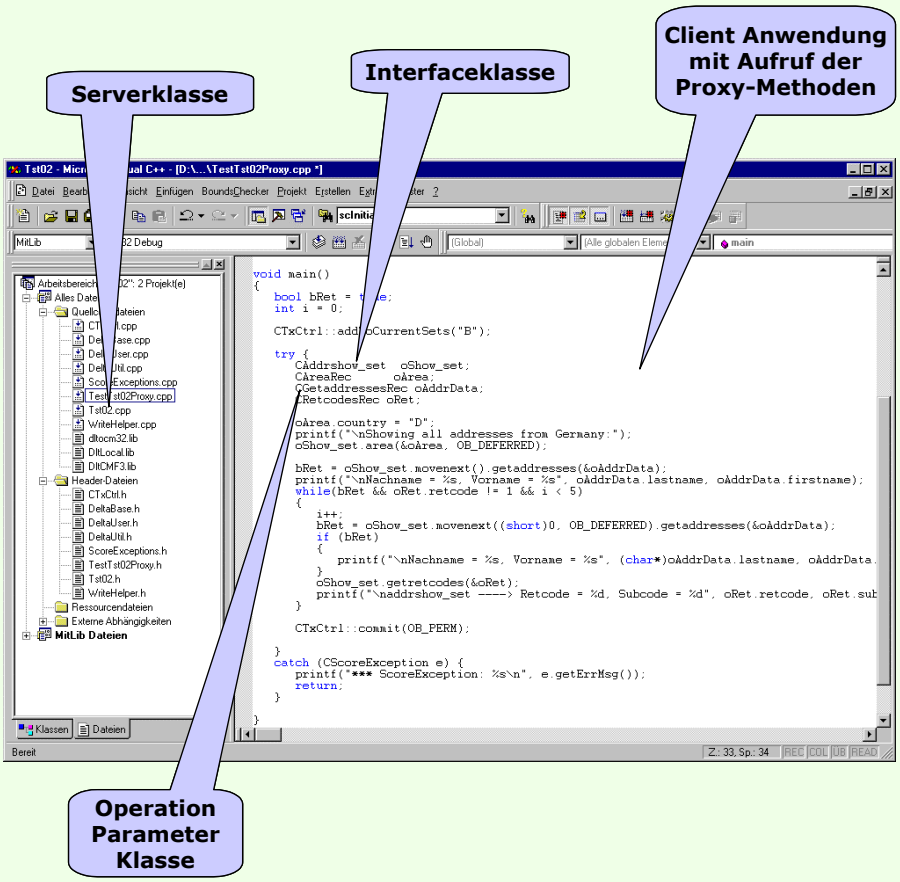
Die Client-Anwendung kann anschließend mit Hilfe der generierten C++-Proxies die Funktionen des Servers über die Methoden dieser Proxies aufrufen.

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. Für den Client-Programmierer ist dieser Vorgang völlig transparent, d.h. es sind keine Änderungen in seiner Anwendung nötig.

GENERIERTE KLASSENTYPEN

Der ObjectBridge-Generator erzeugt zu jedem Component Repository einer Server-Komponente folgende C++ Klassen:



Beispiel für eine mit Hilfe von ObjectBridge für C++ entwickelte Client Anwendung in der Microsoft Visual C++ Umgebung

DIE SERVERKLASSE

Pro Serverkomponente wird eine Serverklasse generiert. Diese bündelt alle C++ 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 C++ Klasse erzeugt. Diese Klasse enthält Methoden für alle Operationen, die das Interface zur Verfügung stellt.

DIE OPERATIONPARAMETER-KLASSE

Für jeden aus einer Struktur bestehenden Parameter einer Operation erzeugt der Generator eine eigene Klasse (Operation Parameter Klasse). Diese C++ Klasse enthält alle Data Elements des Operation Parameters als Zeichenketten-Variablen, die automatisch in die Datentypen der Serverkomponente konvertiert werden und umgekehrt.

SYSTEMVORAUSSETZUNGEN

...FÜR DIE GENERIERUNGSUMGEBUNG

Hardware	IBM-kompabibler PC mit Prozessor der Pentium-Klasse 64MB Hauptspeicher (minimal)
Betriebssysteme	Microsoft Windows@ 98/NT/2000/XP
Software	SCORE Integration Siute ab Version 1.5 SCOUT²

...FÜR DIE LAUFZEITUMGEBUNG

Betriebssysteme	Microsoft Windows@ 98/NT/2000/XP Sun Solaris@ weitere Unix Systeme auf Anfrage
Software	MicrosoftVisual C++ ab Version 5 oder IBM VisualAge C++ ab Version 4.0

MA 13'924.10

www.d-s-t-g.com