

Cold Fusion Application-Server

Genereller Überblick

Eine der Hauptanwendungen von Cold Fusion ist, dass es eine Plattform zur Integration von Datenbanken in Internet und Intranets/Extranets darstellt.

Cold Fusion besitzt eine seitenbasierende Entwicklungsarchitektur und eine leistungsstarke Server-Site Markup Language (CFML = **C**old **F**usion **M**arkup **L**anguage). Dabei ist anzumerken, dass die Entwicklung der CFML-Tags ohne Perl-Scripts auskommt. Mit den mitgelieferten APIs werden Datenbank-Abfragen aus Standard ODBC-Datenbanken wie Access, Foxpro, dBase oder SQL-Server, Oracle und Sybase heraus vorgenommen. Gleichzeitig können Daten aus Formularen von üblichen Browsern eingelesen und an eine Datenbank weitergeleitet werden. Mittels eines bestimmten Tags (<cfmail>) lassen sich z.B. E-Mails empfangen und versenden. Dadurch ist die Integration in HTML und bereits existierender Web Technologien gewährleistet. Dem Cold Fusion Server anhänglich sind umfangreiche Beispiele, die sich für die Entwicklung von Templates verwenden lassen. Für den anspruchsvollen Programmierer stehen ca. 100 zusätzliche Befehle für die Bearbeitung von Variablen, Listen, String, Datei- und Feldbearbeitungs-Methoden zur Verfügung. Das Einbinden von Javascript, Java-Applets und Active-X ist jederzeit möglich. Mittels des Tags CFFILE sind Zugriffe auf einen WWW oder FTP-Server wie Copy, Read, Write, Move, Append und Delete Files verfügbar. Der integrierte Report-Writer (Crystal Reports) ermöglicht die Ausgabe von Berichten oder Formularen auf Windows-Drucker. Cold Fusion greift auf Standarddatenbanken zu, liest nach bestimmten Kriterien Daten aus und liefert via HTTP dem User eine "HTML"-Seite zurück (hierzu siehe auch "Arbeitsprinzip"). Dabei ist die Grundstruktur immer gleich. Dynamische Seiten enthalten meist sehr wenig vorgefertigten Text. Anstelle dessen holen Sie sich die gewünschten Informationen aus anderen Applikationen oder aus Datenbanken. Mögliche Anwendungsfälle sind hierzu Zugriff auf Directories, Visualisierung von Unternehmensdaten oder E-Commerce-Systeme, um nur einige zu nennen.

Überblick über die Cold Fusion Systemkomponenten

Cold Fusion Server: Der Cold Fusion Server stellt das eigentliche Herzstück der Systemkomponenten dar. Er ist ein hochperformanter, gut skalierbarer Applikationsserver, der auf den Plattformen Windows NT und UNIX-Solaris (in Kürze auch unter HP-UX und LINUX) verfügbar ist.

Cold Fusion Studio ist eine Entwicklungsumgebung zur schnellen Entwicklung von Cold Fusion Templates. Auch er besitzt eine direkte Connectivity zu den ODBC-Datenbanken.

Der **Cold Fusion Administrator:** (siehe Bild 1) ist eine Management Console zur Einstellung und Konfigurierung des Servers, die komplett über den WWW-Browser zu bedienen ist.

Die **Cold Fusion Extensions (CFX)** sind ein offenes, XML-basiertes Framework mit dem Cold Fusion in der Lage ist, neuere Serverkomponenten (COM, CORBA, C/C++, VBScript, JavaScript) mit einzubinden.

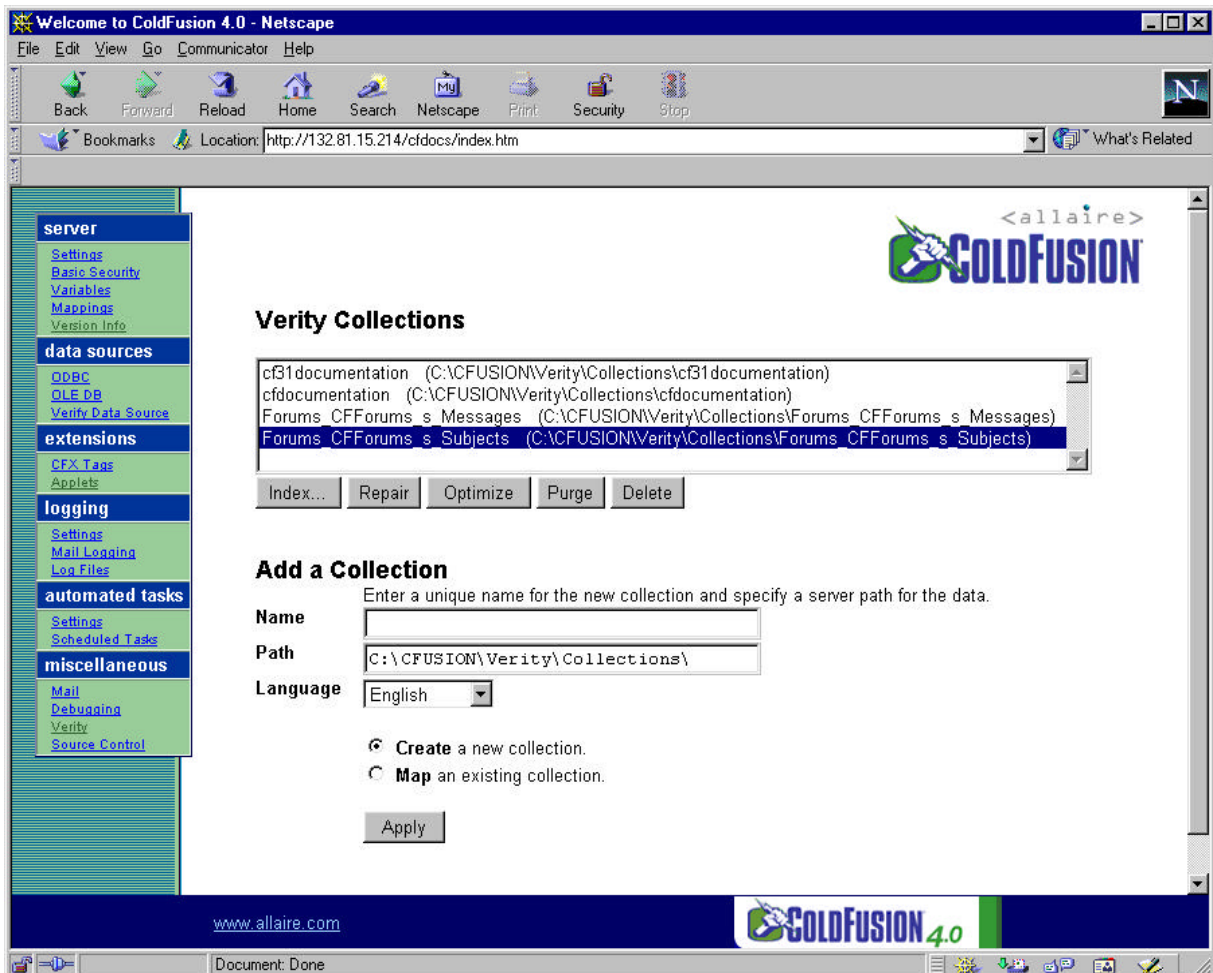


Bild 1: Eine der Cold Fusion Administrator-Seiten

Systemvoraussetzungen

Als Server Plattformen kommen im Moment Windows NT 4.0 oder SUN Solaris in Frage. Auf einer NT-Plattform sollte mindestens 64 MB RAM besser jedoch 128 MB verfügbar sein. Als WWW-Server sind via API momentan der Netscape Server (NSAPI),

der Microsoft Internet Information Server (ISAPI), der O'Reilly WebSite Server (WSAPI) und der Apache API 1.3.2 oder höher verfügbar. Aber auch andere Server sind über das Common Gateway Interface (CGI) mit eingeschränkter Funktionalität bedienbar. Der Cold Fusion Server kann direkt über einen Browser administriert werden. Der Browser sollte jedoch Java 1.01 verstehen und Frames darstellen können. Die Applikationen selber verlangen keinen bestimmten WWW-Browser.

Arbeitsprinzip des Cold Fusion Server

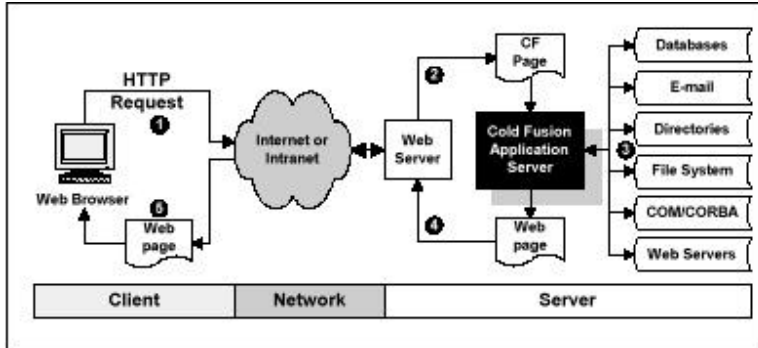


Bild 2: Überblick über das Arbeitsprinzip des Cold Fusion Server

Fast alle Anfragen die aus einem cfm-file heraus generiert werden funktionieren auf folgende Weise: Klickt ein Benutzer im Internet oder Intranet den "Submit-Button" - z.B. in einem Formular den Hypertextlink - an, dann wird ein **HTTP Request** via Intranet oder Internet zum WWW-Server geschickt. (siehe Schritt 1, Bild 2). Der WWW-Server gibt daraufhin die Daten, die er vom Client bekommt an den Applikationsserver über die Server-API weiter. (Schritt 2)

Cold Fusion liest die Daten des Clients aus und arbeitet die CFML-Syntax ab. Abhängig von den jeweiligen CFML-Befehlen interagiert der Server mit DB-Servern, Directory-Systemen, SMTP-Mail-Servern oder anderen Applikationen mittels der CF-API. (Schritt 3). Cold Fusion generiert dynamisch eine HTML-Seite, die an den WWW-Server weitergegeben wird. (Schritt 4). Der WWW-Server gibt die HTML-Seite an den Browser des Clients zurück. (Schritt 5).

Allgemeines über die CFML-Syntax

Die Schnelligkeit und Flexibilität des Allaire Cold Fusion Konzepts liegt in seinen zahllosen Tags, die jeweils für bestimmte Funktionen eingesetzt werden können. Die Tags sind zwar nicht standardisiert, jedoch sehr flexibel einsetzbar und leicht zu erlernen. Den Kern bilden dabei sogenannte Embedded SQL-Befehle mit den SQL-Standard-Kommandos: INSERT INTO, UPDATE, SELECT FROM, DELETE mit denen auf die oben angesprochenen Datenbanken zugegriffen werden kann. Im folgenden ist ein Codelisting zur Veranschaulichung der Syntax abgedruckt:

Listing 1: CFML-Code

```
1: <CFQUERY NAME="Produktliste" DATASOURCE="Fruits_DB">
2: SELECT * FROM Produkte
3: </CFQUERY>
4: <HTML>
5: <HEAD>
6: <TITLE>Produkt- und Preisliste</TITLE>
7: </HEAD>
8: <BODY BGCOLOR="#FFFFFF">
9: <H1>Alle Produkte der Firma Fruits</H1>
```

```
10: <CFOUTPUT QUERY="Produktliste">
11: #Produktname# DM #Preis#<BR>
12: </CFOUTPUT>
13: </BODY>
14: </HTML>
```

Als HTML-Code dieses Beispiels wird dem Browser folgender Code zurückgeliefert

Listing 2: Ergebnis des Listing 1 in HTML:

```
1: <HTML>
2: <HEAD>
3: <TITLE>Produktliste</TITLE>
4: </HEAD>
5: <BODY>
6: <H1>Alle Produkte der Firma Fruits</H1>
7: Äpfel DM 5.00<BR>
8: Birnen DM 7.00<BR>
9: Mangos DM 4.00<BR>
10: Bananen DM 4.50<BR>
11: </BODY>
12: </HTML>
```

In den Zeilen 1 und 2 passiert folgendes: Es wird eine Abfrage (Query-> daher cfquery) mit dem Namen Produktliste gestartet. Dabei wird die Datenbank (datasource) Fruits_DB via ODBC abgefragt. In Zeile 2 wird definiert, welche Tabelle der Datenbank, nämlich die Tabelle Produkte angesprochen werden soll. Wird SELECT * wie in Zeile 2 verwendet, so werden alle Datenbankfelder aus der Tabelle ausgelesen. (Hier ist es auch möglich nur einige ganz gezielt auszulesen).

In Zeile 11 und 12 werden die Daten mittels des Befehls (Tags) <cfoutput> ausgegeben und auf der HTML-Seite dargestellt.

Die Zeilen 7-10 des Listing 2 verdeutlichen, daß alle Werte der DB aus den Feldern (hier im Beispiel sind es zwei) an den Cold Fusion Server zurückgegeben werden.

Die **Cold Fusion Tags (CF-Tags** wie z.B.: <cfquery ...>, s.o.) werden dabei vom Applikationsserver abgearbeitet und der Browser bekommt Standard HTML-Syntax geliefert. Cold Fusion beherrscht dabei mehr als 60 Tags und ca. 200 Funktionen, die auf den Tags basieren. Es lassen sich die unterschiedlichsten Anwendungsfälle damit abdecken. Sollte dieses nicht ausreichen, kann sich der Anwender seine eigenen sogenannten CFX-Tags z.B. mittels C++ schreiben und über den Cold Fusion Server einbinden.

Wichtige Tags mit bestimmten kleinen nützlichen Leistungsmerkmalen sind z.B. die folgenden:

<cffile> Ermöglicht die Dateiverwaltung auf FTP/HTTP-Servern

<cfmail> Ermöglicht das Verschicken von E-Mails

<cfhttp> Lädt Seiten von WWW-Servern und bindet sie z.B. auf der eigenen WWW-Site ein

<cfldap> Greift via LDAP auf X.500 Directory Server zu

<cfsearch> Ermöglicht eine Volltextsuche in Verity Kollektionen (Verity wird als QEM-Engine mit dem Server zusammen ausgeliefert)

<cfpop> Ermöglicht den Zugang auf POP3-Email Postfächer

<cfinclude> Andere Templates können hierdurch in einem Template eingebunden werden

Ein wichtiges interessantes Feature wird dabei durch den <cfmail> Tag realisiert. Damit lassen sich sogar Newsletter- oder Listserverfunktionen nachbilden. Einen Einblick in die Arbeitsweise dieses Tags enthält das Listing 3:

Listing 3: Beispiel für den Tag <cfmail>

```
1: <!-- Senden einer Mail via Mailinglist -->
2: <CFQUERY NAME="GetList" DATASOURCE="Newsletter">
3: SELECT * FROM E_Mail_Tabelle
4: <CFMAIL QUERY="GetList"
5: TO="#Email#"
6: FROM="info@it-fokus.de"
7: SUBJECT="Ihr täglicher Newsletter">
8: Sehr geehrte Damen und Herren...
9: </CFMAIL>
```

In den Zeilen 2 und 3 werden wieder die Daten "eingesammelt" und in den Zeilen 4 bis 9 an alle User, die in der Tabelle E_Mail_Tabelle stehen geschickt. In Zeile 5 wird dafür gesorgt, daß das "TO"- Feld mit Inhalten aus der DB versorgt wird. Die Anzahl der User, die die obige Mail bekommen richtet sich dabei nach der Zahl der DB-Einträge.

Mögliche Anwendungsfälle

Die hauptsächlichen Anwendungsfälle für Cold Fusion Applikationen lassen sich in die folgenden drei Kategorien gliedern:

- **Geschäftsinformationssysteme**
Hier lassen sich z.B. Projektinformationssysteme, Personalmanagementsysteme, elektronische Umfragen mit Online verfügbaren Ein- und Ausgabesystemen oder Lieferverfolgungssysteme aufbauen.
- **E-Commerce-Systeme**
Business-to-Business oder Business-to-Consumer WWW-Applikationen kombinieren nützliche Informationen oder den Service personalisierter Informationen und die Möglichkeit des Produkt- und Dienstleistungsverkaufs via Web. Ein Beispiel für eine gut gemachte E-Commerce Website mit Cold Fusion stellt z.B. Autobytel (<http://www.autobytel.com>) dar.
- **Dynamisches Publizieren von Inhalten**
Hierdurch wird es möglich, Inhalte im Web zu Organisieren, zu Managen und dynamisch zu Verteilen.

Fazit

Cold Fusion hat sich innerhalb weniger Jahre zu einem sehr leistungsfähigen Tool zur Dynamisierung von Intranet und Internet-WWW-Sites entwickelt. Hinsichtlich einiger spezieller Features ist es mittlerweile auch seinen Kinderschuhen entschlüpft. Vor allem das in den in den Vorgängerversionen 2 und 3/3.1 recht schlecht implementierte Fehlerdebugging funktioniert in der aktuellen Version recht gut. Bis auf einige kleinere Bugs auch im Hinblick auf das Zusammenspiel mit der Entwicklungsumgebung Cold Fusion Studio ist das Tool sehr zu empfehlen. Auch der Zugriff auf vorgefertigte Templates mit Hilfe von Wizards hat sich in der Version 4.0 verbessert. So bleiben auf der Wunschliste nur noch einige wenige Features:

- Modularisierung und Bündlung von vorgefertigten Standard-Templates in Form von kompletten Produkten zur Verkürzung von Entwicklungszeiten (in Planung: Codename "Tempest" -> vsl. Erscheinungstermin 2. Hälfte 99)
- Grafisches User-Interface zur objektorientierten Entwicklung ähnlich objektorientierten HTML-Editoren zum einfachen Erstellen und Korrigieren von CFML-Code ohne die Codesprache selbst erlernen zu müssen.

Ressourcen für die Entwicklung von Cold Fusion Applikationen

- a) Allaire -> "Maker" von Cold Fusion (<http://www.allaire.com>)
- b) CFAdvisor -> Cold Fusion Magazin (<http://www.cfadvisor.com>)
- c) Allaire Foren -> Userforen bei Problemen (<http://forums.allaire.com/>)
- d) Ben Forta's Buch -> "The Cold Fusion Web Application Construction Kit" (ISBN: 0-7897-1414-0)
- e) Cold Fusion User Group -> User Group in Zentraleuropa (<http://cfug.interlake.net/>)
- f) Allaire -> Entwicklungsressourcen (<http://www.allaire.com/developer/ReferenceDesk/>)