

FileMaker® Server 11

Custom Web Publishing
mit XML und XSLT



© 2007-2010 FileMaker, Inc. Alle Rechte vorbehalten.

FileMaker, Inc.

5201 Patrick Henry Drive

Santa Clara, California 95054, USA

FileMaker ist eine Marke von FileMaker, Inc., eingetragen in den USA und anderen Ländern. Das Dateiordner-Logo ist eine Marke von FileMaker, Inc. Alle anderen Marken sind Eigentum der jeweiligen Besitzer.

Die FileMaker-Dokumentation ist urheberrechtlich geschützt. Sie dürfen diese Dokumentation ohne schriftliche Genehmigung von FileMaker weder vervielfältigen noch verteilen. Diese Dokumentation darf ausschließlich mit einer gültigen, lizenzierten Kopie der FileMaker-Software verwendet werden.

Alle in den Beispielen erwähnten Personen, Firmen, E-Mail-Adressen und URLs sind rein fiktiv und jegliche Ähnlichkeit mit bestehenden Personen, Firmen, E-Mail-Adressen und URLs ist rein zufällig. Die Danksagungen und Urheberrechtshinweise finden Sie im entsprechenden Dokument, das mit der Software geliefert wurde. Die Erwähnung von Produkten und URLs Dritter dient nur zur Information und stellt keine Empfehlung dar. FileMaker, Inc. übernimmt keine Verantwortung für die Leistung dieser Produkte.

Weitere Informationen finden Sie auf unserer Website unter www.filemaker.de.

Edition: 01

Inhalt

Über dieses Handbuch	9
Kapitel 1	
<i>Einführung in Custom Web Publishing</i>	
Über die Web Publishing Engine	12
Verarbeitung einer Web Publishing Engine -Anforderung	12
Custom Web Publishing mit PHP	13
Custom Web Publishing mit XML und XSLT	13
Vergleich von PHP mit XML und XSLT	14
Gründe für PHP	14
Gründe für XML und XSLT	14
Kapitel 2	
<i>Über Custom Web Publishing mit XML und XSLT</i>	
Erstellen dynamischer Websites mit der Web Publishing Engine	17
Erläuterung von Custom Web Publishing mit XML	17
Erläuterung von Custom Web Publishing mit XSLT	18
Über das Entwickeln von FileMaker XSLT-Stylesheets	18
Wichtige Funktionen in Custom Web Publishing mit XML und XSLT	18
Voraussetzungen für Web Publishing	19
Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing	19
Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung	19
Verbindung zum Internet oder zu einem Intranet	20
Die nächsten Schritte	20
Kapitel 3	
<i>Vorbereiten von Datenbanken für Custom Web Publishing</i>	
Aktivieren von Instant Web Publishing in einer Datenbank	21
Zugriff auf eine geschützte Datenbank	21
Schützen veröffentlichter Datenbanken	22
Web Server-Unterstützung für Internet-Medientypen (MIME)	23
Veröffentlichen des Inhalts von Medienfeldern im Web	23
Veröffentlichen von Medienfeldobjekten, die in einer Datenbank gespeichert sind	23
Veröffentlichen von Medienfeldobjekten, die als Dateiverweis gespeichert sind	23
Anzeige von Medienfelddaten durch Web-Benutzer	24
FileMaker-Scripts und Custom Web Publishing	24
Tipps und Überlegungen zu Scripts	24
Scriptverhalten in Custom Web Publishing-Lösungen	26
Script-Trigger in Custom Web Publishing-Lösungen	26

Kapitel 4

Einführung in Custom Web Publishing mit XSLT

Erläuterung von FileMaker XSLT-Stylesheets	27
Beispiele für die Verwendung von FileMaker XSLT-Stylesheets	27
Einstieg in die Verwendung von Custom Web Publishing mit XSLT	28
Wie die Web Publishing Engine auf der Basis von XML-Daten und XSLT-Stylesheets Seiten generiert	28
Allgemeine Schritte bei der Verwendung von Custom Web Publishing mit XSLT	29
Generieren von FileMaker-XSLT-Stylesheets mithilfe des FileMaker XSLT-Site-Assistenten	30
Vor der Verwendung des XSLT-Site-Assistenten	30
Starten des XSLT-Site-Assistenten	31
Verwenden des XSLT-Site-Assistenten	31
Über die vom XSLT-Site-Assistenten generierten Stylesheets	32
Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm	32
Fehlerbehebung in XSLT-Stylesheets	33

Kapitel 5

Zugriff auf XML-Daten mit der Web Publishing Engine

Verwenden von Custom Web Publishing mit XML	35
Unterschiede zwischen der Web Publishing Engine und FileMaker Pro XML-Import/Export	35
Wie die Web Publishing Engine aus einer Abfrage XML-Daten generiert	36
Allgemeines Verfahren beim Zugriff auf XML-Daten von der Web Publishing Engine	37
Erläuterung der URL-Syntax für XML-Daten und Medienobjekte	38
Erläuterung der URL-Syntax für XML-Daten	38
Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen	38
Erläuterung von URL-Textkodierung	39
Zugriff auf XML-Daten über die Web Publishing Engine	40
Erläuterung von Namensräumen für FileMaker XML	40
Erläuterung der FileMaker-Datenbankfehlercodes	41
Abrufen der Document Type Definitions für die FileMaker-Grammatiken	41
Verwenden der fmresultset-Grammatik	41
Beschreibung der Elemente der fmresultset-Grammatik	42
Beispiel für XML-Daten in der fmresultset-Grammatik	44
Verwenden anderer FileMaker XML-Grammatiken	45
Beschreibung der Elemente in der FMPXMLRESULT-Grammatik	45
Beispiel für XML-Daten in der FMPXMLRESULT-Grammatik	46
Beschreibung der Elemente der Grammatik FMPXMLLAYOUT	46
XML-Daten in der FMPXMLLAYOUT-Grammatik – Beispiel	49
Erläuterung von UTF-8-kodierten Daten	50
Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings	50
Wechseln des Layouts für eine XML-Antwort	52

Erläuterung der Verarbeitung einer XML-Abfrage	53
Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung	53
Fehlerbehebung beim XML-Dokumentzugriff	54

Kapitel 6

Entwickeln von FileMaker XSLT-Stylesheets

Verwenden von XSLT-Stylesheets mit der Web Publishing Engine	55
Referenz der FileMaker XSLT-Erweiterungsfunktionen	56
Über die FileMaker XSLT-Starter-Lösung	56
Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets	56
Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen	57
Verwenden von Query-Strings in FileMaker XSLT-Stylesheets	58
Angeben einer XML-Grammatik für ein FileMaker XSLT-Stylesheet	58
Erläuterung von Namespaces und Präfixen für FileMaker XSLT-Stylesheets	59
Verwenden von statisch definierten Query-Befehlen und Query-Parametern	59
Festlegen von Textkodierung für Abfragen	61
Angeben einer Ausgabemethode und -kodierung	62
Erläuterung der Kodierung von XSLT-Stylesheets	62
Verarbeitung von XSLT-Abfragen, die nicht FileMaker Server abfragen	62
Verwendung von Tokens zur Übergabe von Informationen zwischen Stylesheets	63
Verwenden der FileMaker XSLT-Erweiterungsfunktionen und -parameter	63
Erläuterung der FileMaker-spezifischen XSLT-Parameter, die von der Web Publishing Engine gesetzt werden	64
Verwenden der Query-Information in einer Abfrage	64
Abrufen von Clientinformationen	65
Verwenden des Basis-URI-Parameters der Web Publishing Engine	65
Verwenden des Parameters „authenticated base URI“	66
Laden zusätzlicher Dokumente	66
Verwenden der Layoutinformationen für eine Datenbank in einem Stylesheet	67
Verwenden von Inhaltspufferung	68
Verwenden von Web Publishing Engine-Sitzungen zum Speichern von Informationen zwischen Abfragen	68
Verwenden der Sitzungserweiterungsfunktionen	69
Senden von E-Mail-Nachrichten von der Web Publishing Engine	71
Verwenden von Header-Funktionen	73
Verwenden der Cookie-Erweiterungsfunktionen	74
Verwenden der Erweiterungsfunktionen zur Stringmanipulation	75
Vergleich von Strings mithilfe regulärer Perl 5-Ausdrücke	76
Prüfen von Werten in einem Feld, das als Kontrollkästchen formatiert ist	76
Verwenden der Datums-, Zeit- und Tages-Erweiterungsfunktionen	78
Prüfen des Fehlerstatus von Erweiterungsfunktionen	82

Verwenden der Protokollierung	82
Verwenden der Server-seitigen Verarbeitung von Scriptingsprachen	82
Definieren einer Erweiterungsfunktion	83
Ein Beispiel für eine Erweiterungsfunktion	83

Kapitel 7

Einstellen, Testen und Überwachen einer Site

Einstellen einer Custom Web Publishing-Site	87
Testen einer Custom Web Publishing-Site	88
Beispiele für Stylesheets zum Testen der XML-Ausgabe	89
Überwachen Ihrer Website	90
Verwenden der Web-Server-Zugriffs- und Fehlerprotokolle	90
Verwenden des Web Publishing Engine-Programmprotokolls	90
Verwenden des Web-Server-Modul-Fehlerprotokolls	91
Verwenden von internen Web Publishing Core-Zugriffsprotokollen	91

Anhang A

Gültige Namen in Query-Strings

Erläuterung von Query-Befehlen und -Parametern	93
Richtlinien für die Verwendung von Query-Befehlen und -Parametern	94
Erläuterung der Referenz von FileMaker-Query-Strings	94
Erläuterung der Syntax für einen voll qualifizierten Feldnamen	95
Verwenden von Query-Befehlen für Ausschnittfelder	95
Erläuterung der Syntax für das Angeben eines Variablenfelds	97
Query-Befehlsreferenz	97
Query-Befehl –dbnames (Datenbanknamen)	97
Query-Befehl –delete (Datensatz löschen)	97
Query-Befehl –dup (Datensatz duplizieren)	98
Query-Befehl –edit (Datensatz bearbeiten)	98
Query-Befehle –find, –findall oder –findany (Datensätze suchen)	98
–findquery (Compound-Suche, Query-Befehl)	99
–layoutnames (Layoutnamen) (Query-Befehl)	99
–new (Neuer Datensatz) (Query-Befehl)	99
–process (XSLT-Stylesheets verarbeiten)	100
–scriptnames (Scriptnamen) (Query-Befehl)	100
Query-Befehl –view (Layoutinformationen anzeigen)	100

Query-Parameterreferenz	100
Query-Parameter –db (Datenbankname)	101
–delete.related (Löschen von Ausschnittdatensätzen, Query-Parameter)	101
Query-Parameter –encoding (XSLT-Abfrage-Kodierung)	101
Query-Parameter –field (Medienfeldname)	101
Query-Parameter „fieldname“ (Nicht-Medienfeldname)	102
Query-Parameter „fieldname.Op“ (Vergleichsoperator)	103
Query-Parameter –grammar (Grammatik für XSLT-Stylesheets)	104
Query-Parameter –lay (Layout)	104
Query-Parameter –lay.response (Layout wechseln für Antwort)	104
Query-Parameter –lop (Logischer Operator)	104
Query-Parameter –max (Maximum Datensätze)	105
Query-Parameter –modid (Änderungs-ID)	105
–query (Compound-Suchabfrage, Query-Parameter)	105
Query-Parameter –recid (Datensatz-ID)	106
–relatedsets.filter (Ausschnittdatensätze filtern, Query-Parameter)	106
–relatedsets.max (Ausschnittdatensätze beschränken, Query-Parameter)	107
–script (Script) (Query-Parameter)	107
Query-Parameter –script.param (Parameter an Script übergeben)	108
Query-Parameter –script.prefind (Script vor Suchen)	108
Query-Parameter –script.prefind.param (Parameter an Script vor Suchen übergeben)	108
Query-Parameter –script.presort (Script vor Sortieren)	109
Query-Parameter –script.presort.param (Parameter an Script vor Sortieren übergeben)	109
Query-Parameter –skip (Datensätze überspringen)	109
Query-Parameter –sortfield (Sortierfeld)	110
Query-Parameter –sortorder (Sortierfolge)	110
Query-Parameter –stylehref (Stil-Href)	111
Query-Parameter –styletype (Stilart)	111
Query-Parameter –token.[string] (Werte zwischen XSLT-Stylesheets übergeben)	111

Anhang B

Fehlercodes für Custom Web Publishing

Fehlercodenummern für FileMaker-Datenbanken	113
Fehlercodenummern für die Web Publishing Engine	120
Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen	121

Index	123
--------------	-----

Vorwort

Über dieses Handbuch

Dieses Handbuch setzt voraus, dass Sie mit XML und XSLT, der Entwicklung von Websites und dem Einsatz von FileMaker® Pro zur Erstellung von Datenbanken vertraut sind. Sie sollten mit den Grundlagen der Datenbankgestaltung von FileMaker Pro vertraut sein und die Konzepte von Feldern, Beziehungen, Layouts, Ausschnitten und Medienfeldern verstehen. Dieses Handbuch enthält die folgenden Informationen über Custom Web Publishing mit XML und XSLT auf FileMaker Server:

- Voraussetzungen für die Entwicklung einer Custom Web Publishing-Lösung mit XML oder XSLT
- Veröffentlichen Ihrer Datenbanken mit XML oder XSLT
- Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung
- Abruf von XML-Daten aus Datenbanken, die FileMaker Server bereitstellt
- Entwicklung von FileMaker XSLT-Stylesheets

Wichtig Sie können PDF-Dateien der FileMaker-Dokumentation unter <http://www.filemaker.de/documentation> herunterladen. Aktualisierungen dieses Dokuments erhalten Sie ebenfalls auf der Website.

Die Dokumentation für FileMaker Server umfasst die folgenden Informationen:

Weitere Informationen über	Siehe
Installieren und Konfigurieren von FileMaker Server	<i>FileMaker Server Einführung</i> <i>FileMaker Server Hilfe</i>
Instant Web Publishing	<i>FileMaker Instant Publishing Handbuch</i>
Custom Web Publishing mit PHP	<i>FileMaker Server Custom Web Publishing mit PHP</i>
Verwenden des PHP-Site-Assistenten	<i>PHP-Site-Assistent Hilfe</i>
Custom Web Publishing mit XML und XSLT	<i>FileMaker Server Custom Web Publishing mit XML und XSLT</i> (dieses Handbuch)
Verwenden des XSLT-Site-Assistenten	<i>XSLT-Site-Assistent Hilfe</i>
Installation und Konfiguration der ODBC- und JDBC-Treiber und Verwendung von ODBC und JDBC	<i>FileMaker ODBC- und JDBC-Handbuch</i>
Wie FileMaker Server Auto Update das aktuellste Plugin für FileMaker Pro-Datenbank-Clientcomputer herunterladen kann	<i>FileMaker-Handbuch zur Aktualisierung von Plugins</i>

Kapitel 1

Einführung in Custom Web Publishing

Mit FileMaker Server können Sie Ihre FileMaker-Datenbank im Internet oder in einem Intranet wie folgt veröffentlichen:

Instant Web Publishing: Mit Instant Web Publishing können Sie Ihre Datenbank schnell und einfach im Web veröffentlichen. Sie müssen Ihre Datenbankdateien dazu nicht verändern oder zusätzliche Software installieren – jeder Benutzer mit einem kompatiblen Webbrowser und Zugriff auf das Internet bzw. ein Intranet kann eine Verbindung zu Ihrer Datenbank herstellen, um Datensätze anzuzeigen, zu bearbeiten, zu sortieren oder zu durchsuchen, wenn Sie ihm die entsprechenden Zugriffsrechte geben.

Für Instant Web Publishing muss der Host-Computer FileMaker Pro oder FileMaker Server ausführen. Die Benutzeroberfläche ähnelt der FileMaker Pro-Desktop-Anwendung. Die Web-Seiten und Formulare, mit denen der Web-Benutzer arbeitet, hängen von den in der FileMaker Pro-Datenbank definierten Layouts und Ansichten ab. Weitere Informationen finden Sie im FileMaker Instant Web Publishing Handbuch.

Statisches Veröffentlichen: Wenn sich Ihre Daten selten ändern oder Sie keine Live-Verbindung Ihrer Benutzer zu Ihrer Datenbank wünschen, können Sie die Daten statisch veröffentlichen. Bei der statischen Veröffentlichung exportieren Sie Daten aus einer FileMaker Pro-Datenbank, um eine Web-Seite zu erstellen, die Sie anhand von HTML weiter anpassen können. Die Web-Seite ändert sich nicht, wenn sich Informationen in Ihrer Datenbank ändern, und die Benutzer stellen keine Verbindung zu Ihrer Datenbank her. (Mit Instant Web Publishing werden die Daten in einem Webbrowser-Fenster jedes Mal aktiviert, wenn der Browser eine Anfrage an FileMaker Server sendet.) Weitere Informationen finden Sie im *FileMaker Instant Web Publishing Handbuch*.

Custom Web Publishing: Erscheinungsbild und Funktionalität Ihrer veröffentlichten Datenbank können Sie über die für FileMaker Server verfügbaren Technologien für Custom Web Publishing beeinflussen. Für FileMaker Server, der die veröffentlichten Datenbanken bereitstellt, muss weder FileMaker Pro installiert noch Custom Web Publishing verfügbar sein.

Mit Custom Web Publishing können Sie:

- Ihre Datenbank in andere Web-Sites integrieren.
- festlegen, wie Benutzer mit den Daten arbeiten.
- steuern, wie Daten in Webbrowsern angezeigt werden.

FileMaker Server stellt zwei Custom Web Publishing-Technologien zur Verfügung:

- Custom Web Publishing mit PHP: Verwenden Sie das FileMaker API for PHP, das eine objektorientierte PHP-Schnittstelle zu FileMaker Pro-Datenbanken bereitstellt, um Ihre FileMaker-Daten in einer PHP-Web-Anwendung zu integrieren. Sie können den PHP-Site-Assistenten verwenden, um eine komplette PHP-Website zu generieren, oder Ihre PHP-Web-Seiten selbst programmieren.
- Custom Web Publishing mit XML und XSLT:
 - Verwenden Sie XML Data Publishing, um FileMaker-Daten mit anderen Websites und Anwendungen auszutauschen.
 - Verwenden Sie Server-verarbeitete XSLT-Stylesheets, um beliebige Untergruppen von FileMaker-Daten in andere Websites, mit anderer Middleware und mit Spezialprogrammen zu integrieren. Sie können den XSLT-Site-Assistenten verwenden, um XSLT-Stylesheets zu generieren oder die Stylesheets selbst zu erstellen.

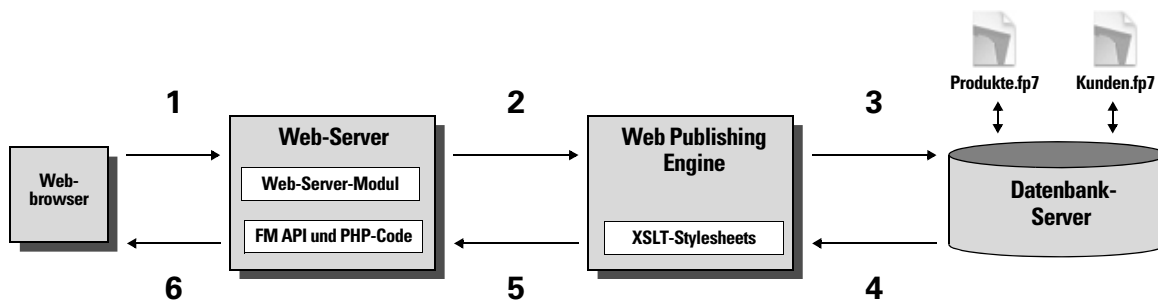
Über die Web Publishing Engine

Um Instant Web Publishing und Custom Web Publishing zu unterstützen, verwendet FileMaker Server einen Satz von Softwarekomponenten, die sogenannte *FileMaker Server Web Publishing Engine*. Die Web Publishing Engine verwaltet Interaktionen zwischen dem Browser eines Web-Benutzers, Ihrem Web-Server und FileMaker Server.

Custom Web Publishing mit XML und XSLT: Die Web Publishing Engine fungiert als XSLT-Prozessor und bietet Ausgaben als HTML, XML oder Text (z. B. vCards) auf dem Web-Server, der dann die Ausgabe für den Webbrowser liefert. Web-Benutzer greifen auf Ihre Custom Web Publishing-Lösung zu, indem sie auf einen HREF-Link klicken oder einen Uniform Resource Locator (URL) eingeben, der die Web-Server-Adresse und eine FileMaker-Query-Zeichenfolgenabfrage angibt. Der URL kann auf XML-Daten zugreifen oder auf ein XSLT-Stylesheet verweisen. Die Web Publishing Engine gibt die XML-Daten, die in der Query-Zeichenfolgenabfrage angegeben sind, oder das Ergebnis des referenzierten XSLT-Stylesheets zurück.

Custom Web Publishing mit PHP: Wenn ein Web-Benutzer auf Ihre Custom Web Publishing-Lösung zugreift, stellt PHP auf FileMaker Server eine Verbindung mit der Web Publishing Engine her und reagiert über das FileMaker API for PHP.

Verwenden der FileMaker Server Web Publishing Engine für Custom Web Publishing



Verarbeitung einer Web Publishing Engine -Anforderung

1. Eine Anforderung wird von einem Webbrowser oder Programm an den Web-Server gesendet.
2. Der Web-Server leitet die Anforderung über das FileMaker Web-Server-Modul an die Web Publishing Engine weiter.
3. Die Web Publishing Engine fordert Daten von der Datenbank an, die der Datenbank-Server bereitstellt.
4. FileMaker Server sendet die angeforderten FileMaker-Daten an die Web Publishing Engine.

5. Die Web Publishing Engine konvertiert die FileMaker-Daten, um die Anforderung zu beantworten.
 - Für PHP-Anforderungen antwortet die Web Publishing Engine auf die API-Anforderung.
 - Für XML-Anforderungen sendet die Web Publishing Engine die XML-Daten direkt an den Web-Server.
 - Für XSLT-Abfragen verwendet die Web Publishing Engine ein XSLT-Stylesheet, um die XML-Daten zu formatieren oder zu transformieren, und generiert die Ausgabe an den Web-Server als HTML-Seiten, XML-Dokument oder Text.
6. Der Web-Server sendet die Ausgabe an den anfordernden Webbrowser oder das anfordernde Programm.

Wichtig Sicherheitsüberlegungen sind wichtig, wenn Sie Daten im Web veröffentlichen. Sehen Sie sich die Sicherheitsrichtlinien im *FileMaker Pro Benutzerhandbuch* an, das als PDF-Datei unter www.filemaker.de/documentation verfügbar ist.

Custom Web Publishing mit PHP

Das FileMaker API for PHP bietet eine objektorientierte PHP-Schnittstelle zu FileMaker-Datenbanken. Mithilfe des FileMaker API for PHP kann sowohl auf Daten als auch auf Logik, die in einer FileMaker Pro-Datenbank gespeichert sind, zugegriffen werden. Diese können auch im Internet veröffentlicht oder an andere Anwendungen exportiert werden. Das API unterstützt zudem komplexe und zusammengesetzte Suchabfragen für das Extrahieren und Filtern von in FileMaker Pro-Datenbanken gespeicherten Daten.

Ursprünglich als prozedurale Programmiersprache entwickelt, wurde PHP als objektorientierte Programmiersprache für die Web-Entwicklung ausgebaut. PHP bietet Programmiersprachenfunktionalität für den Aufbau praktisch jeder Art von Logik innerhalb einer Site-Seite. Zum Beispiel können Sie bedingte Logikkonstrukte verwenden, um die Seitengenerierung, das Datenrouting oder den Workflow zu steuern. PHP bietet zudem Funktionen für Site-Administration und Sicherheit.

Zusätzlich können Sie den FileMaker PHP-Site-Assistenten verwenden, um PHP-Code zu erstellen, der alle erforderlichen Voraussetzungen und Funktionen beinhaltet, um korrekt auf die Daten in einer FileMaker Pro-Datenbank zuzugreifen. Der PHP-Site-Assistent generiert eine mehrseitige Website, die Web-Benutzer in die Lage versetzt, in einer Datenbank zu suchen, eine Liste von Datensätzen anzuzeigen, Datensätze zu durchblättern, zu sortieren, hinzuzufügen, zu bearbeiten, zu duplizieren, zu löschen und einen Statistikbericht anzuzeigen. FileMaker-Entwickler, die nur über geringe PHP-Erfahrung verfügen, können den PHP-Site-Assistenten verwenden, um eine ganze PHP-Website zu generieren. PHP-Entwickler, die nur über geringe Erfahrung mit FileMaker verfügen, können den PHP-Site-Assistenten verwenden, um die Objekte und Methoden des FileMaker API for PHP zu verstehen.

Custom Web Publishing mit XML und XSLT

FileMaker Custom Web Publishing mit XML ermöglicht Ihnen, Abfrageanforderungen an eine FileMaker Pro-Datenbank zu senden, die von FileMaker Server bereitgestellt wird, und die resultierenden Daten anzuzeigen und zu ändern. Mithilfe einer HTTP-Abfrage mit den geeigneten Query-Befehlen und -Parametern können Sie FileMaker-Daten als ein XML-Dokument abrufen. Sie können die XML-Daten dann in andere Programme exportieren oder ihnen ein XSLT-Stylesheet zuweisen.

FileMaker Custom Web Publishing mit XSLT ermöglicht das Umwandeln, Filtern oder Formatieren von XML-Daten für die Verwendung in Webbrowsern oder anderen Programmen. Sie können:

- ein XSLT-Stylesheet verwenden, um die Daten zwischen einer FileMaker XML-Grammatik und einer anderen XML-Grammatik für andere Programme oder andere Datenbanken umzuwandeln.

- die Daten filtern, indem Sie steuern, welche Datenbankfelder vom Stylesheet veröffentlicht werden.
- das Erscheinungsbild der Daten in einer Web-Seite formatieren und steuern, wie der Web-Benutzer mit den Daten interagiert.

Die Web Publishing Engine verwendet Ihre Stylesheets, um Daten aus einer FileMaker-Datenbank zu beziehen, wenn ein Web-Benutzer eine HTTP-Abfrage und einen URL sendet, der auf eines Ihrer XSLT-Stylesheets verweist. Die Web Publishing Engine verwendet ein Stylesheet, um die XML-Daten umzuwandeln und zu formatieren, und generiert die resultierende HTML-Seite, mit der der Web-Benutzer arbeiten kann.

Der FileMaker XSLT-Site-Assistent ist eine Anwendung, mit der Sie grundlegende XSLT-Stylesheets als Ausgangspunkt für Custom Web Publishing mit XSLT erstellen können. Der XSLT-Site-Assistent generiert Stylesheets für Seiten, die die Datenbank durchsuchen, einen Datensatz anzeigen, die Datensätze in der Datenbank auflisten, Datensätze sortieren, hinzuzufügen, bearbeiten, duplizieren, löschen und einen Statistikbericht anzeigen.

Vergleich von PHP mit XML und XSLT

Der folgende Abschnitt zeigt einige Richtlinien auf, um die beste Lösung für Ihre Site zu bestimmen.

Gründe für PHP

- PHP ist eine leistungsstarke, objektorientierte prozedurale Scriptingsprache und relativ leicht zu erlernen. Es gibt viele Ressourcen für Schulung, Entwicklung und Support.
- Mithilfe des FileMaker API for PHP kann sowohl auf Daten als auch auf Logik, die in einer FileMaker Pro-Datenbank gespeichert sind, zugegriffen werden. Diese können auch im Internet veröffentlicht oder an andere Anwendungen exportiert werden.
- Mit PHP können Sie bedingte Logik verwenden, um den Seitenaufbau oder den Fluss zu kontrollieren.
- PHP bietet Programmiersprachenfunktionalität für den Aufbau vieler Arten von Logik innerhalb einer Site-Seite.
- PHP ist eine der beliebtesten Web-Scriptingsprachen.
- PHP ist eine Open-Source-Sprache, verfügbar unter <http://php.net>.
- PHP ermöglicht den Zugriff auf eine große Vielzahl von Third-Party-Komponenten, die Sie in Ihre Lösungen integrieren können.

Hinweis Weitere Informationen über Custom Web Publishing mit PHP finden Sie im Handbuch *FileMaker Server Custom Web Publishing mit PHP*.

Gründe für XML und XSLT

- Die Syntax der FileMaker XML-Abfrageparameter ist für die Datenbankinteraktion konzipiert und vereinfacht die Lösungsentwicklung.
- XML und XSLT sind W3C-Standards.
- XML ist ein von Maschinen und Menschen lesbares Format, das Unicode unterstützt, so dass Daten in jeder geschriebenen Sprache kommuniziert werden können.
- XML ist bestens geeignet für die Darstellung von Datensätzen, Listen und Daten mit Baumstruktur.
- Mit XSLT können Sie die XML-Ausgabe in strukturierte Textdokumente wie RSS, RTF oder vCard umwandeln.

- Sie können XSLT verwenden, um die XML-Ausgabe aus einer Grammatik in eine andere umzuwandeln.
- Vorlagen vereinfachen die Anwendung bedingter Formatierung auf variable Daten.
- Sie können FMPXMLRESULT-basierte Stylesheets für Custom Web Publishing und für den XML-Export aus FileMaker Pro-Datenbanken verwenden.
- FileMaker Server verwaltet die Verarbeitung von FileMaker-XSLT-Stylesheets und verhindert den unbefugten Zugriff auf Daten, die bei Verwendung von Client-seitigen XSLT-Stylesheets ungesichert sein könnten.

Kapitel 2

Über Custom Web Publishing mit XML und XSLT

Erstellen dynamischer Websites mit der Web Publishing Engine

Die Web Publishing Engine bietet Custom Web Publishing für FileMaker Server mithilfe von XML-Datenveröffentlichung und vom Server verarbeiteten XSLT-Stylesheets. Custom Web Publishing bietet mehrere Vorteile:

- **Anpassung:** Sie können festlegen, wie Web-Benutzer mit FileMaker-Daten interagieren und wie die Daten in Webbrowsern angezeigt werden.
- **Datenaustausch:** Mithilfe von FileMaker XML können Sie FileMaker-Daten mit anderen Websites und Programmen austauschen.
- **Datenintegration:** Mit FileMaker XSLT-Stylesheets können Sie FileMaker-Daten in andere Websites und mit anderen Middleware- und Spezialprogrammen integrieren. Sie können die Daten so darstellen, als würden sie zu einer anderen Website gehören, anstatt ein vollständiges FileMaker-Layout im Webbrowser anzuzeigen.
- **Sicherheit:** Der FileMaker Server Administrator kann Instant Web Publishing, XML Web Publishing oder XSLT Web Publishing einzeln für alle Datenbanken, die der Server bereitstellt, aktivieren oder deaktivieren. Als Eigentümer der FileMaker-Datenbank können Sie den Zugriff der Web-Benutzer auf Instant Web Publishing, XML Web Publishing oder XSLT Web Publishing für jede Datenbank steuern.
- **Server-seitige Stylesheets:** Server-seitige Verarbeitung von XSLT-Stylesheets verhindert unbefugte Anzeige von vertraulicher Datenbankinformation, die mit Client-seitigen Stylesheets möglich sein könnte.
- **Steuern und Filtern von veröffentlichten Daten:** Mit XSLT-Stylesheets können Sie die Daten und die Art der Datenbankinformationen, die Sie veröffentlichen wollen, steuern und filtern und damit unbefugte Nutzung der Datenbank verhindern. Sie können auch Metadaten ausblenden wie z. B. Datenbank- und Feldnamen.
- **Basierend auf offenen Standards:** Ihnen stehen mehr Tools, Ressourcen und Fachpersonal für Custom Web Publishing-Lösungen zur Verfügung. Wenn Sie Standard-XML oder -XSLT kennen, können Sie sofort mit der Entwicklung von Lösungen beginnen, nachdem Sie ein paar spezielle Details über Custom Web Publishing mit XML gelernt haben, z. B. die erforderliche URL-Syntax und Query-Parameter.

Erläuterung von Custom Web Publishing mit XML

Mit Custom Web Publishing mit XML können Sie Daten aus FileMaker-Datenbanken abrufen und die Daten einfach in anderen Ausgabeformaten verwenden. Mithilfe einer HTTP-Abfrage mit den geeigneten Query-Befehlen und -Parametern können Sie FileMaker-Daten als ein XML-Dokument abrufen. Sie können die XML-Daten dann in anderen Programmen verwenden oder ihnen ein XSLT-Stylesheet zuweisen. Weitere Informationen finden Sie unter Kapitel 5, „Zugriff auf XML-Daten mit der Web Publishing Engine“.

Erläuterung von Custom Web Publishing mit XSLT

Custom Web Publishing mit XSLT ermöglicht das Umwandeln, Filtern oder Formatieren von XML-Daten für die Verwendung in einem Webbrowser oder in anderen Programmen. Sie können ein XSLT-Stylesheet verwenden, um die Daten zwischen einer FileMaker XML-Grammatik und einer anderen XML-Grammatik für den Einsatz in einem anderen Programm oder einer anderen Datenbank umzuwandeln. Sie können die Daten filtern, indem Sie steuern, welche Datenbankfelder vom Stylesheet veröffentlicht werden. Sie können das Erscheinungsbild der Daten in einer Web-Seite formatieren und steuern, wie der Web-Benutzer mit den Daten interagiert. Weitere Informationen finden Sie unter Kapitel 4, „Einführung in Custom Web Publishing mit XSLT“.

Die Web Publishing Engine verwendet Ihre Stylesheets, um Daten dynamisch aus einer FileMaker-Datenbank zu beziehen, wenn ein Web-Benutzer eine HTTP-Abfrage und einen URL sendet, der auf eines Ihrer XSLT-Stylesheets verweist. Die Web Publishing Engine verwendet ein Stylesheet, um die XML-Daten umzuwandeln und zu formatieren, und generiert die resultierende HTML-Seite, mit der der Web-Benutzer arbeiten kann.

Besuchen Sie für zusätzliche Informationen zur Verwendung von FileMaker Server Custom Web Publishing mit XML und XSLT die Web-Site www.filemaker.de/documentation.

Über das Entwickeln von FileMaker XSLT-Stylesheets

FileMaker Server beinhaltet ein Tool zur Entwicklung von XSLT-Stylesheets. Der FileMaker XSLT-Site-Assistent ist eine Anwendung, mit der Sie grundlegende XSLT-Stylesheets als Ausgangspunkt für Custom Web Publishing mit XSLT erstellen können. Der XSLT-Site-Assistent eignet sich ideal, um den Aufbau von FileMaker XSLT-Stylesheets kennenzulernen. Sie können dann Ihr eigenes Erstellungswerkzeug für XSLT-Stylesheets verwenden, um die Stylesheets je nach Bedarf zu ändern. Weitere Informationen finden Sie unter „Generieren von FileMaker-XSLT-Stylesheets mithilfe des FileMaker XSLT-Site-Assistenten“ auf Seite 30.

Hinweis FileMaker Server unterstützt XSLT 1.0 wie durch das World Wide Web Consortium definiert. Von Ihnen verwendete XSLT-Authoring-Tools müssen den Standards entsprechendes XSLT 1.0 erzeugen.

Wichtige Funktionen in Custom Web Publishing mit XML und XSLT

FileMaker Server Custom Web Publishing mit XML und XSLT bietet mehrere wichtige neue Funktionen:

- Datenbanken werden auf FileMaker Server bereitgestellt und FileMaker Pro muss nicht ausgeführt werden.
- Sie können Server-seitige XSLT-Stylesheet-Verarbeitung nutzen, die mehr Sicherheit als die Client-seitige Stylesheet-Verarbeitung bietet.
- Sie können die Server-seitige Verarbeitung von JavaScript in XSLT-Stylesheets nutzen. Weitere Informationen finden Sie unter „Verwenden der Server-seitigen Verarbeitung von Scriptingsprachen“ auf Seite 82.
- Sie können mit Ihrem FileMaker XSLT-Stylesheet unbefugte Verwendung von Query-Befehlen und Query-Parametern verhindern, indem Sie die zu verwendenden Query-Befehle, Parameter und Werte bei der Anforderung von XML-Daten statisch definieren. Weitere Informationen finden Sie unter „Verwenden von statisch definierten Query-Befehlen und Query-Parametern“ auf Seite 59.
- Wie in FileMaker Pro basiert der Zugriff auf Daten, Layouts und Felder auf den Benutzerkontoeinstellungen, die in den Zugriffsrechten der Datenbank definiert sind. Die Web Publishing Engine unterstützt auch mehrere andere Sicherheitsverbesserungen. Weitere Informationen finden Sie unter „Schützen veröffentlichter Datenbanken“ auf Seite 22.
- Web-Benutzer können komplexe Scripts mit mehreren Schritten ausführen. FileMaker unterstützt ungefähr 70 Scriptschritte in Custom Web Publishing. Informationen finden Sie im Abschnitt „FileMaker-Scripts und Custom Web Publishing“ auf Seite 24.

- Sie können einen Parameterwert an ein FileMaker-Script übergeben. Weitere Informationen finden Sie unter „Query-Parameter –script.param (Parameter an Script übergeben)“ auf Seite 108, „Query-Parameter –script.pfind.param (Parameter an Script vor Suchen übergeben)“ auf Seite 108 und „Query-Parameter –script.presort.param (Parameter an Script vor Sortieren übergeben)“ auf Seite 109.
- Mit der fmresultset-XML-Grammatik können Sie nach Name auf Felder zugreifen und relatedset- (Ausschnitt) Daten manipulieren.
- Mithilfe von Sitzungsfunktionen in einem XSLT-Stylesheet können Sie die Informationen und Transaktionen eines Web-Benutzers in vom Server geführten Sitzungen speichern.
- Für den Zugriff auf Daten in einer Datenbank müssen Sie ein Layout angeben. Weitere Informationen finden Sie unter Anhang A, „Gültige Namen in Query-Strings“.
- Jeder Web-Benutzer kann über einen eindeutigen Variablenfeldwert verfügen, der für die Dauer der Sitzung bestehen bleibt. Allgemeine Informationen zu Variablenfeldern finden Sie in der FileMaker Pro Hilfe. Informationen über die Verwendung von Variablenfeldern mit Custom Web Publishing finden Sie unter „Erläuterung der Syntax für das Angeben eines Variablenfelds“ auf Seite 97.

Voraussetzungen für Web Publishing

Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing

Um Datenbanken mithilfe von Custom Web Publishing mit XML oder XSLT zu veröffentlichen, benötigen Sie:

- einen FileMaker Server-Einsatz, der Folgendes umfasst:
 - einen Web-Server, entweder Microsoft IIS (Windows) oder Apache (Mac OS X)
 - den FileMaker Datenbank-Server, aktiviert für Custom Web Publishing
 - die Web Publishing Engine, installiert und konfiguriert
- eine oder mehrere FileMaker Pro-Datenbanken, die FileMaker Server bereitstellt
- IP-Adresse oder Domänenname des Hosts, auf dem der Web-Server läuft
- einen beliebigen Webbrowser und Zugriff auf den Web-Server, um Ihre Custom Web Publishing-Lösung zu entwickeln und zu testen

Weitere Informationen finden Sie im Handbuch *FileMaker Pro Einführung*.

Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung

Für den Zugriff auf eine Custom Web Publishing-Lösung mit XML oder XSLT benötigen Web-Benutzer:

- einen Webbrowser
- Zugang zum Internet oder zu einem Intranet und zum Web-Server
- IP-Adresse oder Domänenname des Hosts, auf dem der Web-Server läuft

Wenn die Datenbank passwortgeschützt ist, müssen Web-Benutzer auch einen Benutzernamen und ein Passwort für ein Datenbankkonto eingeben.

Verbindung zum Internet oder zu einem Intranet

Wenn Sie Datenbanken im Internet oder in einem Intranet veröffentlichen, muss auf dem Host-Computer FileMaker Server laufen und die freizugebenden Datenbanken müssen bereitgestellt und verfügbar sein. Zusätzlich:

- Veröffentlichen Sie Ihre Datenbank auf einem Computer, der über eine ständige Verbindung zum Internet bzw. Intranet verfügt. Sie können Datenbanken zwar ohne ständige Verbindung veröffentlichen, aber sie stehen Web-Benutzern nur zur Verfügung, wenn Ihr Computer mit dem Internet bzw. Intranet verbunden ist.
- Der Host-Computer für den Web-Server, der Teil des FileMaker Server-Einsatzes ist, muss über eine eigene *statische* (permanente) IP-Adresse oder einen Domänennamen verfügen. Wenn Sie die Verbindung zum Internet über einen Internet-Anbieter (ISP) herstellen, könnte Ihre IP-Adresse *dynamisch zugewiesen* werden (d. h., sie ändert sich bei jeder Verbindung). Mit einer dynamischen IP-Adresse ist es für Web-Benutzer schwieriger, Ihre Datenbanken zu finden. Wenn Sie sich über Ihre Zugangsart nicht sicher sind, wenden Sie sich an Ihren ISP oder Netzwerkadministrator.

Die nächsten Schritte

Hier einige Vorschläge, um mit der Entwicklung von Custom Web Publishing-Lösungen zu beginnen:

- Falls noch nicht geschehen, aktivieren Sie Custom Web Publishing mithilfe der FileMaker Server Admin Console. Informationen hierzu finden Sie in der FileMaker Server Hilfe und im Handbuch *FileMaker Server Einführung*.
- Öffnen Sie in FileMaker Pro jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank die geeigneten erweiterten Zugriffsrechte für Custom Web Publishing aktiviert sind. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 21.
- Wie Sie mithilfe von XML auf Daten in FileMaker-Datenbanken zugreifen, erfahren Sie in Kapitel 5, „Zugriff auf XML-Daten mit der Web Publishing Engine“.
- Wie Sie mit der Entwicklung von FileMaker XSLT-Stylesheets beginnen, lesen Sie in Kapitel 4, „Einführung in Custom Web Publishing mit XSLT“.

Kapitel 3

Vorbereiten von Datenbanken für Custom Web Publishing

Bevor Sie Custom Web Publishing mit einer Datenbank verwenden können, müssen Sie die Datenbank vorbereiten und vor unbefugtem Zugriff schützen.

Aktivieren von Instant Web Publishing in einer Datenbank

Sie müssen Custom Web Publishing in jeder Datenbank aktivieren, die Sie veröffentlichen wollen. Sie können Custom Web Publishing mit XML oder Custom Web Publishing mit XSLT in jeder Datenbank einzeln oder beide Technologien gleichzeitig aktivieren. Wenn Sie in einer Datenbank keine dieser Technologien aktivieren, können Web-Benutzer Custom Web Publishing nicht für den Zugriff auf die Datenbank verwenden, selbst wenn sie von FileMaker Server bereitgestellt wird, der zur Unterstützung einer Web Publishing Engine konfiguriert ist.

So aktivieren Sie Custom Web Publishing in einer Datenbank:

1. Öffnen Sie in FileMaker Pro die zu veröffentlichende Datenbank mit einem Konto, das über die Berechtigung für vollen Zugriff verfügt. Alternativ können Sie die Datenbank mit einem Konto öffnen, das über die Berechtigung „Erweiterte Zugriffsrechte verwalten“ verfügt.
2. Weisen Sie einer oder mehreren Berechtigungen eine oder beide dieser erweiterten Zugriffsrechte zu:
 - Verwenden Sie für Custom Web Publishing mit XML dieses Schlüsselwort: `fmxml`
 - Verwenden Sie für Custom Web Publishing mit XSLT dieses Schlüsselwort: `fmxmlt`Seit FileMaker Pro Version 8 sind die Schlüsselwörter `fmxml` und `fmxmlt` im Register „Erweiterte Zugriffsrechte“ für Sie definiert.
3. Weisen Sie die Berechtigung(en), die die erweiterten Custom Web Publishing-Zugriffsrechte enthalten, einem oder mehreren Konten bzw. dem Admin- oder Gastkonto zu.

Hinweis Wenn Sie Kontonamen und Passwörter für Custom Web Publishing-Lösungen definieren, verwenden Sie druckbare ASCII-Zeichen. Beispiel: a-z, A-Z und 0-9. Verwenden Sie für sicherere Kontonamen und Passwörter zudem Satzzeichen wie „!“ und „%“, aber verwenden Sie keine Doppelpunkte. Informationen über das Einrichten von Konten finden Sie in der FileMaker Pro Hilfe.

Zugriff auf eine geschützte Datenbank

Wenn Web-Benutzer eine Custom Web Publishing-Lösung für den Zugriff auf eine Datenbank verwenden, werden sie eventuell aufgefordert, ihre Kontoinformationen einzugeben. Wenn das Gastkonto für die Datenbank deaktiviert ist oder über keine Berechtigung mit dem erweiterten Zugriffsrecht für Custom Web Publishing verfügt, verwendet die Web Publishing Engine die HTTP-Standardauthentifizierung, um entsprechende Informationen von Web-Benutzern zu verlangen. Der Browser des Web-Benutzers zeigt das Dialogfeld für die HTTP-Standardauthentifizierung an, in dem der Benutzer einen Benutzernamen und ein Passwort für ein Konto eingibt, das über ein erweitertes Zugriffsrecht für Custom Web Publishing verfügt.

Die folgende Liste fasst die Abläufe zusammen, wenn ein Web-Benutzer eine Custom Web Publishing-Lösung für den Zugriff auf eine Datenbank verwendet:

- Wenn Sie einem Konto kein Passwort zugewiesen haben, geben Web-Benutzer nur den Kontonamen an.
- Wenn das Gastkonto deaktiviert ist, werden die Benutzer beim Zugriff auf die Datenbank aufgefordert, Kontoname und Passwort anzugeben. Für das Konto muss ein erweitertes Zugriffsrecht für Custom Web Publishing aktiviert sein.
- Wenn das Gastkonto aktiviert ist und über eine Berechtigung mit einem erweiterten Zugriffsrecht für Custom Web Publishing verfügt, öffnen automatisch alle Web-Benutzer die Datenbank mit den Zugriffsrechten, die dem Gastkonto zugewiesen sind. Wenn das erweiterte Zugriffsrecht für Custom Web Publishing dem Gastkonto zugewiesen ist:
 - Web-Benutzer werden nicht aufgefordert, Kontoname und Passwort einzugeben, wenn sie eine Datei öffnen.
 - Alle Web-Benutzer werden automatisch mit dem Gastkonto angemeldet und erhalten die Gast-Zugriffsrechte. Sie können Benutzern über den Scriptschritt „Erneut anmelden“ erlauben, ihre Anmeldekonto von einem Webbrowser aus zu wechseln (z. B. vom Gastkonto zu einem anderen Konto mit mehr Rechten).
 - Die Standardberechtigungen für Gastkonten umfassen den Nur-Lese-Zugriff. Sie können für dieses Konto die Standardberechtigungen einschließlich erweiterten Zugriffsrechten ändern. Weitere Informationen finden Sie in der FileMaker Pro Hilfe.

Hinweis Standardmäßig können Web-Benutzer ihr Kontopasswort nicht von einem Webbrowser aus ändern. Sie können diese Funktion über den Scriptschritt „Passwort ändern“ in Ihre Datenbank integrieren, damit Web-Benutzer ihr Passwort vom Browser aus ändern können. Weitere Informationen finden Sie in der FileMaker Pro Hilfe.

Schützen veröffentlichter Datenbanken

Wenn Sie Custom Web Publishing mit XML oder XSLT verwenden, können Sie beschränken, wer auf Ihre veröffentlichten Datenbanken zugreifen kann.

- Weisen Sie Datenbankkonten, die für Custom Web Publishing verwendet werden, Passwörter zu.
- Aktivieren Sie Custom Web Publishing mit XML oder XSLT nur in den Berechtigungen für Konten, denen Sie den Zugriff auf Ihre veröffentlichten Datenbanken erlauben wollen.
- Um einen Typ der Custom Web Publishing-Technologie für eine einzelne Datenbank zu aktivieren oder zu deaktivieren, wählen Sie das entsprechende erweiterte Zugriffsrecht.
- Aktivieren oder deaktivieren Sie einen Typ der Custom Web Publishing-Technologie für alle Custom Web Publishing-Lösungen in der Web Publishing Engine mit der FileMaker Server Admin Console. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

- Konfigurieren Sie Ihren Web-Server, um die IP-Adressen einzuschränken, die über die Web Publishing Engine auf Ihre Datenbanken zugreifen können. Beispielsweise können Sie angeben, dass nur Web-Benutzer von der IP-Adresse 192.168.100.101 auf Ihre Datenbanken zugreifen können. Informationen über das Einschränken von IP-Adressen finden Sie in der Dokumentation zu Ihrem Web-Server.
- Verwenden Sie Secure Sockets Layer- (SSL) Verschlüsselung für die Kommunikation zwischen Ihrem Web-Server und den Browsern der Web-Benutzer. Die SSL-Verschlüsselung wandelt Informationen, die zwischen Servern und Clients übertragen werden, mithilfe von mathematischen Formeln in unverständliche Informationen um. Der englische Fachausdruck für diese Chiffrier-Algorithmus ist Ciphers. Diese Algorithmen nutzt der Empfänger, um mithilfe von Schlüsseln, den sogenannten Chiffrierschlüsseln, die Informationen wieder in verständliche Daten umzuwandeln. Informationen über das Aktivieren und Konfigurieren von SSL erhalten Sie in der Dokumentation zu Ihrem Web-Server.

Weitere Informationen zur Sicherung Ihrer Datenbank finden Sie im *FileMaker Pro Benutzerhandbuch*, das als PDF-Datei unter www.filemaker.de/documentation verfügbar ist.

Web Server-Unterstützung für Internet-Medientypen (MIME)

Ihr Web-Server bestimmt die Unterstützung für die aktuellen MIME-Typen (MIME; Multipurpose Internet Mail Extensions), die für das Internet registriert sind. Die Web Publishing Engine ändert die MIME-Unterstützung eines Web-Servers nicht. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Web-Server.

Veröffentlichen des Inhalts von Medienfeldern im Web

Der Inhalt eines Medienfelds wie z. B. eine Bilddatei kann in einer FileMaker-Datenbank oder als Dateiverweis mit einem relativen Pfad gespeichert werden.

Hinweis Die Web Publishing Engine unterstützt kein Streaming von Filmen. Web-Benutzer müssen eine ganze Filmdatei herunterladen, bevor sie den Film ansehen können.

Veröffentlichen von Medienfeldobjekten, die in einer Datenbank gespeichert sind

Wenn ein Medienfeld die tatsächlichen Dateien in der FileMaker-Datenbank speichert, ist keine Aktion mit dem Inhalt des Medienfelds erforderlich, sofern die Datenbankdatei korrekt auf dem FileMaker Server freigegeben und zugänglich ist. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen“ auf Seite 38, und „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen“ auf Seite 57.

Veröffentlichen von Medienfeldobjekten, die als Dateiverweis gespeichert sind

Wenn ein Medienfeld Dateiverweise anstelle tatsächlicher Dateien speichert, müssen Sie folgende Schritte ausführen, um die Medienfeldobjekte zu veröffentlichen.

Hinweis Alle QuickTime-Filme werden als Verweis in einem Medienfeld gespeichert.

So veröffentlichen Sie Medienfeldobjekte, die als Dateiverweise gespeichert sind:

1. Speichern Sie die Medienobjektdateien im Web-Ordner des FileMaker Pro-Ordners.
2. Fügen Sie in FileMaker Pro die Objekte in das Medienfeld ein und wählen Sie die Option Nur Verweis auf die Datei speichern.

3. Kopieren oder verschieben Sie die Objektdateien, auf die verwiesen wird, im Web-Ordner an die gleiche Stelle im relativen Pfad des Root-Ordners der Web-Server-Software.

- IIS: Verschieben Sie die Dateien in: <Laufwerk>:\inetpub\wwwroot
- Apache: Verschieben Sie die Dateien in: /Library/WebServer/Documents

Hinweis Für Medienobjekte, die als Dateiverweise gespeichert sind, muss Ihr Web-Server so konfiguriert sein, dass er die MIME-Typen für alle Arten von Dateien unterstützt, die Sie anbieten wollen (z. B. Filme). Weitere Informationen finden Sie in der Dokumentation zu Ihrem Web-Server.

Anzeige von Medienfelddaten durch Web-Benutzer

Wenn Sie eine Datenbank mit der Web Publishing Engine im Web veröffentlichen, können Web-Benutzer wie folgt eingeschränkt mit Daten in Medienfeldern arbeiten:

- Web-Benutzer können keine Töne abspielen oder OLE-Objekte in einem Medienfeld anzeigen – stattdessen wird eine Grafik angezeigt.
- Web-Benutzer können den Inhalt eines Medienfelds nicht ändern oder ihm Inhalte hinzufügen. Web-Benutzer können ein Medienfeld nicht verwenden, um Daten in die Datenbank hochzuladen.
- Falls Ihre Datenbank Grafiken enthält, die nicht in GIF- oder JPEG-Format sind, erstellt die Web Publishing Engine ein temporäres JPEG-Bild, wenn die Grafikdaten von einem Webbrowser angefordert werden.

FileMaker-Scripts und Custom Web Publishing

Die Funktion „Scripts verwalten“ in FileMaker Pro kann sich häufig wiederholende Aufgaben automatisieren und mehrere Aufgaben zusammenfassen. In Kombination mit Custom Web Publishing ermöglichen FileMaker-Scripts den Web-Benutzern, mehrere Aufgaben oder eine Reihe von Aufgaben durchzuführen.

FileMaker unterstützt über 75 Scriptschritte in Custom Web Publishing. Web-Benutzer können eine Vielzahl automatisierter Aufgaben ausführen, wenn Sie Scripts in einer Query-Zeichenfolge für einen URL oder in einer <?xslt-cwp-query?>-Verarbeitungsanweisung in einem XSLT-Stylesheet verwenden. Um nicht unterstützte Scriptschritte anzuzeigen, wählen Sie im Dialogfeld „Script bearbeiten“ in FileMaker Pro die Option Web Publishing aus der Liste Kompatibilität anzeigen. Grau dargestellte Scripts werden nicht im Web unterstützt. Informationen über das Erstellen von Scripts finden Sie in der FileMaker Pro Hilfe.

Tipps und Überlegungen zu Scripts

Auch wenn viele Scriptschritte im Web identisch funktionieren, gibt es einige, die sich anders verhalten. Weitere Informationen erhalten Sie unter „Scriptverhalten in Custom Web Publishing-Lösungen“ auf Seite 26. Testen Sie alle Scripts, die von einem Webbrowser ausgeführt werden, bevor Sie Ihre Datenbank bereitstellen. Stellen Sie sicher, dass Sie sich mit unterschiedlichen Benutzerkonten anmelden, damit Sie sichergehen können, dass die Scripts für alle Clients gleichermaßen funktionieren. Prüfen Sie die Protokolldatei der Web Publishing Engine (pe_application_log.txt) auf Scripting-bezogene Fehler. Weitere Informationen finden Sie unter „Verwenden des Web Publishing Engine-Programmprotokolls“ auf Seite 90.

Beachten Sie folgende Tipps und Überlegungen:

- Verwenden Sie Konten und Zugriffsrechte, um die Scripts einzuschränken, die ein Web-Benutzer ausführen kann. Überprüfen Sie, ob die Scripts nur Web-kompatible Scriptschritte enthalten und nur Zugang zu Scripts gewähren, die von einem Webbrowser aus ausgeführt werden sollen.

- Berücksichtigen Sie die Nebeneffekte von Scripts, die eine Reihe von Scriptschritten ausführen, die durch Zugriffsrechte kontrolliert werden. Wenn ein Script beispielsweise einen Scriptschritt zum Löschen von Datensätzen enthält und der Web-Benutzer sich nicht mit einem Konto anmeldet, das das Löschen von Datensätzen zulässt, führt das Script den Scriptschritt zum Löschen von Datensätzen nicht aus. Das Script könnte jedoch weiter ausgeführt werden, so dass unerwartete Ergebnisse auftreten können.
- Wählen Sie im Dialogfeld „Script bearbeiten“ Script mit vollen Zugriffsrechten ausführen, damit Scripts Aufgaben ausführen können, für die Sie dem Benutzer an sich keinen Zugang gewähren wollen. Beispielsweise können Sie Benutzern untersagen, mit ihren Konten und Zugriffsrechten Datensätze zu löschen, ihnen aber die Ausführung eines Scripts gestatten, das unter vordefinierten Bedingungen innerhalb eines Scripts bestimmte Arten von Datensätzen löscht.
- Wenn Ihre Scripts Schritte enthalten, die nicht unterstützt werden (z. B. Schritte, die nicht Web-kompatibel sind), verwenden Sie den Scriptschritt **AnwenderAbbruchZulassen setzen**, um festzulegen, wie nachfolgende Schritte gehandhabt werden.
 - Wenn der Scriptschritt **AnwenderAbbruchZulassen setzen** aktiviert (ein) ist, hindern nicht unterstützte Scriptschritte das Script an der Fortsetzung.
 - Wenn **AnwenderAbbruchZulassen setzen** ausgeschaltet ist, werden nicht unterstützte Scriptschritte übersprungen und das Script setzt die Ausführung fort.
 - Wenn dieser Scriptschritt nicht enthalten ist, werden Scripts ausgeführt, als ob die Funktion aktiv wäre, d. h., nicht unterstützte Scriptschritte stoppen Scripts.
- Einige Scripts, die mit einem Schritt von einem FileMaker Pro-Client aus funktionieren, benötigen eventuell einen zusätzlichen Schritt „Schreibe Änderung Datens./Abfrage“, um die Daten auf dem Host zu speichern. Da Web-Benutzer nicht über eine direkte Verbindung zum Host verfügen, werden sie nicht sofort benachrichtigt, wenn sich Daten ändern. Funktionen wie bedingte Wertelisten funktionieren z. B. nicht gleichermaßen für Web-Benutzer, da die Daten auf dem Host gespeichert werden müssen, bevor sich die Wirkung in einem Wertelistenfeld zeigt.
- Scripts, die Daten verändern, sollten den Schritt „Schreibe Änderung Datens./Abfrage“ enthalten, da Datenänderungen erst im Browser sichtbar werden, wenn die Daten gespeichert bzw. am Server „bestätigt“ wurden. Dies gilt für einige Scriptschritte wie Ausschneiden, Kopieren, Einfügen usw. Viele Aktionen mit nur einem Scriptschritt sollten in Scripts umgewandelt werden, die den Schritt „Schreibe Änderung Datens./Abfrage“ enthalten. Wenn Sie Scripts entwerfen, die von einem Webbrowser aus ausgeführt werden, nehmen Sie den Schritt „Schreibe Änderung Datens./Abfrage“ am Ende des Scripts auf, um sicherzustellen, dass alle Änderungen gespeichert werden.
- Um von der Art des Clients abhängige Scripts zu erstellen, verwenden Sie die Funktion **Hole(ProgrammVersion)**. Wenn der zurückgegebene Wert **Web Publishing Engine** enthält, wissen Sie, dass der aktuelle Benutzer über Custom Web Publishing auf Ihre Datenbank zugreift. Weitere Informationen zu Funktionen finden Sie in der FileMaker Pro Hilfe.
- Wenn Sie ein Script in einem XSLT-Stylesheet benutzen, das einen Status festlegt oder ändert, müssen Sie mithilfe von FileMaker Server Admin Console die Option **XSLT-Datenbanksitzungen für die Web Publishing Engine** aktivieren. Andernfalls werden Status zwischen Abfragen nicht beibehalten. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Scriptverhalten in Custom Web Publishing-Lösungen

Die nachfolgenden Scriptschritte arbeiten im Web anders als in FileMaker Pro. Weitere Informationen zu allen Scriptschritten finden Sie in der FileMaker Pro Hilfe.

Scriptschritt	Verhalten in Custom Web Publishing-Lösungen
Script ausführen	Scripts können nur dann in anderen Dateien ausgeführt werden, wenn die Dateien auf FileMaker Server bereitgestellt werden und Custom Web Publishing in den anderen Dateien aktiviert ist.
Programm beenden	Meldet Web-Benutzer ab, schließt die Fenster, beendet aber den Webbrowser nicht.
AnwenderAbbruchZulassen setzen	Legt fest, wie nicht unterstützte Scriptschritte gehandhabt werden. Aktivieren, damit Scripts gestoppt werden; deaktivieren, damit nicht unterstützte Schritte übersprungen werden. Weitere Informationen finden Sie unter „Tipps und Überlegungen zu Scripts“ auf Seite 24. Web-Benutzer können Custom Web Publishing-Scripts nicht abbrechen, aber diese Option ermöglicht, dass nicht unterstützte Scriptschritte die Fortsetzung des Scripts stoppen.
Fehleraufzeichnung setzen	Diese Option ist bei Custom Web Publishing immer aktiviert. Web-Benutzer können Custom Web Publishing-Scripts nicht abbrechen.
Scriptpause setzen	Dieses Script wird zwar in Custom Web Publishing unterstützt, sollte aber nicht benutzt werden. Wenn der Scriptschritt „Scriptpause setzen“ ausgeführt wird, hält das Script an. Nur ein Script, das erneut den Schritt „Scriptpause setzen“ enthält, kann die Ausführung fortsetzen. Wenn das Script im Pausezustand bleibt, bis die Sitzung endet, wird das Script nicht vollständig ausgeführt.
Datensätze sortieren	Sie müssen eine Sortierfolge mit dem Scriptschritt „Datensätze sortieren“ speichern, damit er in Custom Web Publishing ausgeführt wird.
URL öffnen	Dieser Scriptschritt hat keine Wirkung in einer Custom Web Publishing-Lösung.
Gehe zu Feld	Im Webbrowser können Sie „Gehe zu Feld“ nicht verwenden, um zu einem bestimmten Feld zu wechseln. Sie können diesen Scriptschritt jedoch in Kombination mit anderen Scriptschritten verwenden, um Aufgaben auszuführen. Beispielsweise können Sie zu einem Feld gehen und den Inhalt kopieren und dann zu einem anderen Feld gehen und den Wert einfügen. Um die Wirkung im Browser zu sehen, müssen Sie den Datensatz mit dem Scriptschritt „Datensatz bestätigen“ speichern.
Schreibe Änderung Datens./Abfrage	Sendet den Datensatz zur Datenbank.

Script-Trigger in Custom Web Publishing-Lösungen

In FileMaker Pro können sowohl Scripts als auch Benutzeraktionen (wie zum Beispiel ein Benutzer, der in ein Feld klickt) Script-Trigger aktivieren. In Custom Web Publishing können jedoch nur Scripts Script-Trigger aktivieren. Wenn ein Custom Web Publishing-Benutzer zum Beispiel in ein Feld klickt, das einen Script-Trigger „BeiObjektBetreten“ besitzt, wird der Trigger nicht aktiviert. Wenn ein Script jedoch dazu führt, dass die Einfügemarke in das Feld wechselt, wird der Script-Trigger „BeiObjektBetreten“ aktiviert. Weitere Informationen zu Script-Trigger finden Sie in der FileMaker Pro Hilfe.

Kapitel 4

Einführung in Custom Web Publishing mit XSLT

FileMaker XSLT-Stylesheets ermöglichen das Umwandeln, Filtern oder Formatieren von XML-Daten für die Verwendung in einem Webbrowser oder in anderen Programmen und Anwendungen. Dieses Kapitel stellt FileMaker XSLT-Stylesheets und ein Tool vor, das Ihnen den Einstieg in die Erstellung von XSLT-Stylesheets erleichtert: den FileMaker XSLT-Site-Assistenten. Weitere Informationen über den Aufbau von FileMaker XSLT-Stylesheets finden Sie in Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Erläuterung von FileMaker XSLT-Stylesheets

Mithilfe von FileMaker XSLT-Stylesheets können Sie:

- FileMaker-Daten filtern, indem Sie steuern, welche Datenbankfelder vom Stylesheet veröffentlicht werden.
- Metadaten wie z. B. Datenbank- und Feldnamen ausblenden.
- das Erscheinungsbild der Daten in einer Web-Seite formatieren und steuern, wie der Web-Benutzer mit den Daten interagiert.
- die Daten als HTML oder Text ausgeben, z. B. vCards oder durch Komma getrennte Werte.
- die Daten von einer FileMaker XML-Grammatik in eine andere XML-Grammatik umwandeln, um sie in einer anderen Datenbank oder Anwendung zu nutzen, z. B. Scalable Vector Graphics (SVG).
- eine Untergruppe der FileMaker-Daten in andere Websites und mit anderen Middleware- und Spezialanwendungen integrieren, die sich erheblich von der FileMaker-Datenbank unterscheiden können.
- die veröffentlichten Feldnamen ändern und unbefugte Verwendung der Datenbankdesigninformation verhindern.

Hinweis Custom Web Publishing mit XSLT für FileMaker Server basiert auf der W3C-Empfehlung für XSLT 1.0. Informationen über XSLT 1.0 finden Sie unter www.w3.org. Zusätzliche Funktionalität wie Sitzungsverwaltung, E-Mail-Versand und Zugriff auf Cookies und Header wird durch FileMaker XSLT-Erweiterungsfunktionen zur Verfügung gestellt. Weitere Informationen finden Sie unter „Verwenden der FileMaker XSLT-Erweiterungsfunktionen und -parameter“ auf Seite 63. Die Web Publishing Engine unterstützt nicht XSL Formatting Objects (XSL-FO).

Beispiele für die Verwendung von FileMaker XSLT-Stylesheets

Ein paar Beispiele für die unzähligen Möglichkeiten mit FileMaker XSLT-Stylesheets:

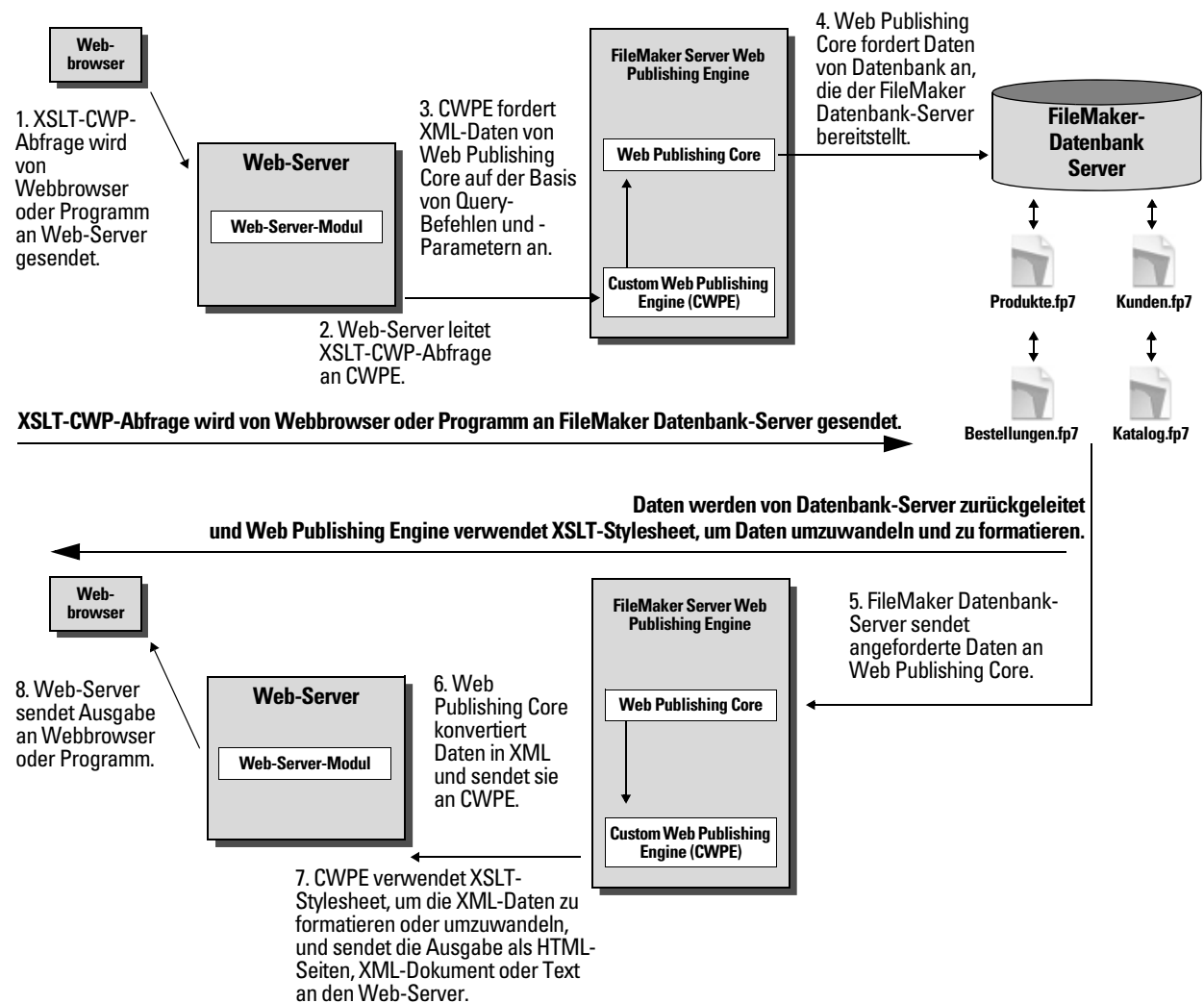
- Sie können eine Tabelle in eine Web-Seite mit einem Teil der Daten aus einer FileMaker-Datenbank einfügen, damit Web-Benutzer darin blättern können. Die Tabelle könnte z. B. Namen und Adressen, aber keine Telefonnummern enthalten. Um unbefugten Zugriff zu verhindern, kann die Web-Seite statt der tatsächlichen Feldnamen in der FileMaker-Datenbank (z. B. „vor_name“) generische Beschriftungen für die Daten zeigen (z. B. „Name“).
- Sie können eine Web-Seite oder eine Anwendung erstellen, die Daten aus einem FileMaker-Ausschnitt in Informationen aus anderen Datenquellen integriert.
- Sie können einer Web-Seite eine Taste hinzufügen, über die eine vCard aus den Kontaktinformationen einer Person in einer FileMaker-Datenbank erzeugt wird.
- Sie können die XML-Daten aus einer FileMaker-Datenbank in eine XML-Grammatik umwandeln, die eine Tabellenkalkulations- oder Datenbankanwendung öffnen kann.

Einstieg in die Verwendung von Custom Web Publishing mit XSLT

Wenn Sie Standard-XML und XSLT kennen, können Sie sofort mit dem Einsatz der Web Publishing Engine beginnen, nachdem Sie ein paar spezielle Details über FileMaker XML und XSLT Publishing wie die Verwendung von FileMaker-XSLT-Erweiterungsfunktionen, Query-Befehlen und Query-Parametern gelernt haben. Der XSLT-Site-Assistent hilft Ihnen bei der Erstellung von Stylesheets und zeigt Ihnen den Aufbau der Stylesheets. Sie können die Stylesheets mithilfe Ihres bevorzugten XML- und XSLT-Erstellungswerkzeugs weiter ausarbeiten.

Wie die Web Publishing Engine auf der Basis von XML-Daten und XSLT-Stylesheets Seiten generiert

Nachdem eine XSLT Custom Web Publishing (XSLT-CWP)-Abfrage an den Web-Server gesendet wurde, fragt die Web Publishing Engine die FileMaker-Datenbank auf der Basis der Query-Befehle und -Parameter ab, die im Stylesheet und im URL definiert sind, und gibt die Daten dann gemäß den Anweisungen im XSLT-Stylesheet aus.



Allgemeine Schritte bei der Verwendung von Custom Web Publishing mit XSLT

Zusammenfassung der Schritte bei der Verwendung von Custom Web Publishing mit XSLT:

1. Stellen Sie in Admin Console sicher, dass XSLT Publishing aktiviert ist. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
2. Öffnen Sie in FileMaker Pro jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank das erweiterte Zugriffsrecht fmxslt für Custom Web Publishing mit XSLT aktiviert ist. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 21.

Hinweis Stellen Sie sicher, dass Sie entsprechende FileMaker-Datenbankberechtigungen verwenden, wenn Sie Stylesheets entwickeln, die an den Endbenutzer gegeben werden. Andernfalls haben Sie in der FileMaker-Datenbank eventuell Zugriff auf Layouts und Funktionen, die dann dem Endbenutzer nicht zur Verfügung stehen und damit zu inkonsistentem Verhalten führen.

3. Erstellen Sie XSLT-Stylesheets, die FileMaker-spezifische XSLT-Erweiterungsfunktionen, Query-Befehle und Query-Parameter enthalten, um die XML-Daten aus einer FileMaker-Datenbank zu formatieren oder umzuwandeln.

Sie können mithilfe des FileMaker XSLT-Site-Assistenten eines oder mehrere grundlegende XSLT-Stylesheets als Ausgangsposition für Ihre Site erstellen. Weitere Informationen finden Sie im nächsten Abschnitt, „Generieren von FileMaker-XSLT-Stylesheets mithilfe des FileMaker XSLT-Site-Assistenten“.

Sie können auch Ihr eigenes XSLT-Erstellungs- oder Textbearbeitungswerkzeug verwenden, um die XSLT-Stylesheets wie erforderlich zu ändern oder Ihre Stylesheets ganz neu zu entwickeln. Weitere Informationen finden Sie unter Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

4. Kopieren oder platzieren Sie die XSLT-Stylesheets im Ordner `xslt-template-files`, der sich im Ordner Web Publishing im Ordner FileMaker Server auf dem Host befindet, auf dem die Web Publishing Engine installiert ist.

Sie können die Stylesheets auch in einem optionalen Ordner oder einer Ordnerhierarchie innerhalb des Ordners `xslt-template-files` platzieren.

5. Platzieren Sie etwaige statische Dateien auf dem Web-Server. Weitere Informationen finden Sie unter „Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm“ auf Seite 32.

6. Erstellen oder ändern Sie eine Website oder ein Programm, das die XSLT-Stylesheets verwendet.

Sie können z. B. für die Website eine statische Seite wie `index.html` verwenden, die Web-Benutzer automatisch an ein XSLT-Stylesheet weiterleitet oder einen Link auf das XSLT-Stylesheet enthält.

7. Richten Sie unbedingt Sicherheitsmechanismen für Ihre Site oder Ihr Programm ein.
8. Testen Sie die Site bzw. das Programm mit den XSLT-Stylesheets unter Verwendung der Konten und Berechtigungen, die für Web-Benutzer definiert sind.
9. Geben Sie die Site bzw. das Programm frei und informieren Sie Benutzer.

Generieren von FileMaker-XSLT-Stylesheets mithilfe des FileMaker XSLT-Site-Assistenten

Der FileMaker XSLT-Site-Assistent ist eine Anwendung, mit der Sie grundlegende XSLT-Stylesheets als Ausgangspunkt für Custom Web Publishing mit XSLT erstellen können. Der XSLT-Site-Assistent eignet sich ideal, um den Aufbau von FileMaker XSLT-Stylesheets kennenzulernen. Sie können dann Ihr eigenes Erstellungswerkzeug für XSLT-Stylesheets oder Textverarbeitungswerkzeuge verwenden, um die Stylesheets je nach Bedarf zu verändern. Sie können den XSLT-Site-Assistenten nicht zur Bearbeitung oder Aktualisierung bestehender Stylesheets verwenden, aber Sie können ihn verwenden, um die anfänglichen Stylesheets für eine gesamte Site zu generieren oder ein einzelnes Stylesheet zu erzeugen und damit einer bestehenden Site zusätzliche Funktionalität (z. B. das Löschen von Datensätzen) zu geben.

Mithilfe des XSLT-Site-Assistenten können Sie XSLT-Stylesheets für alle Arten von Seiten generieren, die nützlich für die Arbeit mit FileMaker-Datenbanken über Custom Web Publishing sind. Abhängig von dem im XSLT-Site-Assistenten gewählten Optionen können Sie eine Site erstellen, die Benutzern das Folgende erlaubt:

- Anzeige einzelner Datensätze
- Anzeige einer Liste mit allen Datensätzen in der Datenbank
- Durchsuchen der Datenbank und Anzeigen des Ergebnisses in einer Liste
- Sortieren von Datensätzen
- Hinzufügen von Datensätzen
- Bearbeiten und Duplizieren von Datensätzen
- Löschen von Datensätzen
- Anzeigen eines Statistikberichts

Ferner können Sie eine optionale Homepage generieren, die mit den anderen generierten XSLT-Stylesheet-Seiten verknüpft ist.

Die Web Publishing Engine verwendet jedes Ihrer Stylesheets, um Daten dynamisch aus einer FileMaker-Datenbank zu beziehen, wenn ein Web-Benutzer eine HTTP-Anforderung und einen URL sendet, der eines Ihrer XSLT-Stylesheets referenziert. Die Web Publishing Engine verwendet ein Stylesheet, um die XML-Daten umzuwandeln und zu formatieren, und generiert die resultierende HTML-Seite, mit der der Web-Benutzer arbeiten kann.

Hinweis Die Site Assistant-Stylesheets transformieren FileMaker XML-Daten basierend auf der XML-Grammatik `fmresultset` in HTML-Seiten, so dass die Stylesheets nicht kompatibel mit anderer Verwendung von XML-Daten wie dem FileMaker Pro XML-Export sind.

Vor der Verwendung des XSLT-Site-Assistenten

Bevor Sie den XSLT-Site-Assistenten verwenden können, um XSLT-Stylesheets für eine Datenbank zu generieren:

- Setzen Sie in der Datenbank das erweiterte Zugriffsrecht `fmxml`. Verwenden Sie bei der Ausführung des XSLT-Site-Assistenten Berechtigungen, die denen entsprechen, die Sie den Web-Benutzern geben. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 21.
- Öffnen Sie die Datenbank auf der Datenbank-Server-Komponente von FileMaker Server und geben Sie sie frei. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

- Stellen Sie sicher, dass die Web-Server-Komponente des FileMaker Server-Einsatzes ausgeführt wird.
- Stellen Sie sicher, dass die Web Publishing Engine-Komponente des FileMaker Server-Einsatzes ausgeführt wird.
- Aktivieren Sie XSLT Publishing in der Web Publishing Engine, um die XSLT-Stylesheets zu verwenden und zu testen. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Starten des XSLT-Site-Assistenten

Hinweis Um den XSLT-Site-Assistenten zu verwenden, muss Java Runtime Environment 5 oder Java Runtime Environment 6 installiert sein.

So starten Sie den XSLT-Site-Assistenten:

1. Öffnen Sie einen Browser mit der Seite FileMaker Server Web Publishing-Tools.

Wechseln Sie zu folgendem URL.

`http://<Server>:16000/tools`

wobei <Server> der Rechner ist, auf dem sich FileMaker Server befindet.

2. Klicken Sie auf PHP-Site-Assistent und XSLT-Site-Assistent, um die Seite FileMaker Server Web Publishing-Tools aufzurufen.

3. Klicken Sie auf XSLT-Site-Assistenten starten.

FileMaker Server installiert die erforderlichen JAR-Dateien auf Ihrem lokalen Rechner und zeigt einen Statusdialog an, bis der Vorgang abgeschlossen ist.

4. (Optional) Nachdem die Dateien installiert sind, können Sie wählen, ob Sie ein Symbol für den XSLT-Site-Assistenten auf Ihrem Desktop installieren möchten. Klicken Sie auf OK, um das Symbol zu installieren.

Sie können jetzt beginnen, den XSLT-Site-Assistenten zu verwenden.

Verwenden des XSLT-Site-Assistenten

Details und umfassende schrittweise Anweisungen für die Verwendung des XSLT-Site-Assistenten finden Sie in der XSLT-Site-Assistent Hilfe. Weitere Informationen über die Verwendung der vom XSLT-Site-Assistenten generierten XSLT-Stylesheets finden Sie unter „Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm“ auf Seite 32.

Wichtig Wenn Sie bei Verwendung des XSLT-Site-Assistenten eine Datenbank wählen, die mehrere Tabellen enthält, wählen Sie unbedingt Layouts, die mit dieser Tabelle verbunden sind, sonst liefert die generierte Site unerwartete Ergebnisse. Eine Datenbank könnte z. B. die Tabelle „Produkte“ und die Tabelle „Kunden“ enthalten. Wenn Sie die Layouts für eine Seite zum Suchen, eine Seite zur Datensatzbearbeitung und eine Seite zum Hinzufügen von Datensätzen wählen, müssen diese Layouts unbedingt mit derselben Tabelle verknüpft sein.

Über die vom XSLT-Site-Assistenten generierten Stylesheets

Die XSLT-Stylesheets, die der XSLT-Site-Assistent generiert, umfassen mehrere FileMaker-spezifische Verarbeitungsanweisungen, Elemente und Parameter. Einige Beispiele dazu:

- Die Verarbeitungsanweisung `<?xslt-cwp-query params="query string-fragment"?>` gibt die zu verwendende XML-Grammatik an und definiert den Namen der im XSLT-Site-Assistenten gewählten Datenbank statisch. Weitere Informationen finden Sie unter „Verwenden von statisch definierten Query-Befehlen und Query-Parametern“ auf Seite 59.
- Das Element `<xsl:param name="request-query"/>` wird verwendet, um auf Query-Informationen in einer Abfrage oder in HTML-Formulardaten zuzugreifen. Mithilfe dieses Elements kann z. B. in den XSLT-Site-Assistent-Stylesheets auf die aktuelle Abfrage-Query-Information zugegriffen werden, um die aktuelle Position in einer Datensatz-Ergebnismenge zu bestimmen und Verknüpfungen zum vorherigen und nächsten Datensatz zu erstellen. Weitere Informationen finden Sie unter „Verwenden der Query-Information in einer Abfrage“ auf Seite 64.
- Das Element `<xsl:param name="authenticated-xml-base-uri"/>`, das nicht immer eingeschlossen ist, wird verwendet, um auf den authentifizierten Basis-URI in einer Abfrage zuzugreifen, wenn in der Abfrage mehr XML-Daten benötigt werden. Weitere Informationen finden Sie unter „Verwenden des Parameters „authenticated base URI““ auf Seite 66.

Der XSLT-Site-Assistent generiert auch das Stylesheet `utilities.xml` zur Definition von Fehlern und gängigen XSLT-Vorlagen, die von mehreren XSLT-Site-Assistent-Stylesheets aufgerufen werden.

Informationen über andere Abschnitte der XSLT-Site-Assistent-Stylesheets finden Sie unter Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm

Unabhängig davon, ob Sie XSLT-Stylesheets mithilfe des XSLT-Site-Assistenten generiert oder von Grund auf neu entwickelt haben, sind die Schritte für die Benutzung in einer Website oder einem Programm mit der Web Publishing Engine gleich.

So verwenden Sie FileMaker XSLT-Stylesheets in einer Website oder einem Programm:

1. Kopieren oder platzieren Sie die XSLT-Stylesheets im Ordner `xslt-template-files`, der sich im Ordner Web Publishing im Ordner FileMaker Server auf dem Host befindet, auf dem die Web Publishing Engine installiert ist.

Sie können die Stylesheets auch in einem optionalen Ordner oder einer Ordnerhierarchie innerhalb des Ordners `xslt-template-files` platzieren.

2. Wenn Ihre XSLT-Stylesheets auf statische Dateien verweisen, z. B. statische Bilder oder HTML-Dateien, platzieren Sie die statischen Dateien mit ihrer originalen Ordnerhierarchie innerhalb des Root-Ordners auf dem Web-Server. Stellen Sie sicher, dass der relative Pfad beibehalten wird.

Nehmen Sie z. B. an, ein XSLT-Stylesheet verweist auf die Bilddatei `logo.jpg` und verwendet dafür das HTML-Tag ``. Die Datei `logo.jpg` muss sich auf dem Web-Server an der folgenden Position befinden:

```
<root folder>/fmi/xsl/logo.jpg
```

3. Falls ein Datenbank-Medienfeld einen Dateiverweis anstelle einer tatsächlichen Datei speichert, muss das entsprechende Medienobjekt im Ordner FileMaker Pro Web gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird, und dann in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software kopiert oder verschoben werden. Weitere Informationen finden Sie unter „Veröffentlichen des Inhalts von Medienfeldern im Web“ auf Seite 23.

Hinweis Falls das Medienfeld die tatsächlichen Dateien in der FileMaker-Datenbank speichert, ist mit dem Inhalt des Medienfelds keine Aktion erforderlich, wenn die Datenbank richtig auf dem FileMaker Server freigegeben und zugänglich ist.

4. Um ein XSLT-Stylesheet abzufragen und zu verarbeiten, verwenden Sie die folgende URL-Syntax:

```
<scheme>://<host>[:<port>]/fmi/xsl/<folder>/<stylesheet>.xsl[?<query string>]
```

Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 56.

Hinweis Für Websites ist es praktisch, ein XSLT-Stylesheet als Homepage zu verwenden, die nicht verlangt, dass Benutzer für den Zugriff einen Query-String eingeben. Der XSLT-Site-Assistent kann eine Datei `home.xsl` erzeugen, die keinen Query-String verlangt, da sie die Verarbeitungsanweisung `<?xslt-cwpp-query?>` verwendet. Wenn Sie z. B. Ihre Stylesheets (einschließlich eines `home.xsl`-Stylesheets) in den Ordner `my_templates` im Ordner `xslt-template-files` kopiert haben, können Sie folgenden URL verwenden, um die Stylesheets anzufordern und zu verarbeiten:

```
http://192.168.123.101/fmi/xsl/my_templates/home.xsl
```

Wichtig Die Web Publishing Engine erlaubt Web-Benutzern nicht, den Quelltext der XSLT-Stylesheets anzuzeigen, die im Ordner `xslt-template-files` installiert sind. Wenn Web-Benutzer eine Anforderung für die Verarbeitung eines Stylesheets senden, sendet die Web Publishing Engine nur das Ergebnis der Stylesheet-Umwandlung an den Webbrowser bzw. das Programm.

Fehlerbehebung in XSLT-Stylesheets

Wenn Probleme beim Verwenden der XSLT-Stylesheets auftreten, überprüfen Sie das Folgende:

- Die erweiterten Zugriffsrechte in der Datenbank müssen für Custom Web Publishing mit XSLT eingestellt und einem Benutzerkonto zugewiesen sein. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 21.
- Die Datenbank ist auf der Datenbank-Server-Komponente von FileMaker Server bereitgestellt und geöffnet. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
- Prüfen Sie, ob der verwendete Datenbankkontoname und ggf. Ihr Passwort korrekt sind.
- Die Web-Server-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- Die Web Publishing Engine-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- XSLT-Publishing muss in der Web Publishing Engine aktiviert sein.
 - Öffnen Sie einen Browser mit der Seite FileMaker Server Technologietests:

```
http://<Server>:16000/test
```

 wobei `<Server>` der Rechner ist, auf dem sich FileMaker Server befindet.
 - Klicken Sie auf den Link XSLT Custom Web Publishing testen, um eine XSLT-Seite zu öffnen, die auf die Testdatenbank `FMServer_Sample` zugreift.

Weitere Informationen finden Sie im Handbuch *FileMaker Server Einführung* und in der FileMaker Server Hilfe.

Kapitel 5

Zugriff auf XML-Daten mit der Web Publishing Engine

Sie können FileMaker-Daten in Extensible Markup Language- (XML) Format mithilfe der Web Publishing Engine beziehen und aktualisieren. So wie HTML zur Standardbeschreibungssprache für die Kommunikation im World Wide Web wurde, hat sich XML zur Standardsprache für den strukturierten Datenaustausch entwickelt. Diese Sprache wird von zahlreichen Einzelpersonen, Organisationen und Unternehmen zur Übermittlung von Produktinformationen, Transaktionen, Inventardaten und anderen geschäftlichen Informationen genutzt.

Verwenden von Custom Web Publishing mit XML

Wenn Sie Standard-XML kennen, können Sie sofort mit dem Einsatz der Web Publishing Engine beginnen, nachdem Sie ein paar spezielle Details über Custom Web Publishing mit XML wie z. B. die erforderliche URL-Syntax und Query-Parameter gelernt haben.

Indem Sie HTTP-URL-Abfragen mit FileMaker-Query-Befehlen und -Parametern verwenden, können Sie eine Datenbank abfragen, die von FileMaker Server bereitgestellt wird, und die resultierenden Daten in XML-Format herunterladen. Sie können eine Datenbank beispielsweise auf alle Datensätze in einem bestimmten Postleitzahlenbereich abfragen und die resultierenden XML-Daten wie gewünscht verwenden.

Sie können auch die Server-seitigen XSLT-Stylesheets der Web Publishing Engine verwenden, um die XML-Daten zu filtern, die Daten in HTML oder Text wie vCards umzuformatieren oder die Daten in andere XML-Grammatiken wie Scalable Vector Graphics (SVG) umzuwandeln. Weitere Informationen finden Sie unter Kapitel 4, „Einführung in Custom Web Publishing mit XSLT“ und Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Allgemeine Informationen zu XML, weitere Beispiele, in denen es verwendet wird, sowie Links zu XML-Ressourcen finden Sie auf der FileMaker-Website unter www.filemaker.de.

Hinweis Die von der Web Publishing Engine generierten XML-Daten sind wohlgeformt und richten sich nach der XML 1.0-Spezifikation. Informationen zu den Anforderungen des XML-Aufbaus finden Sie in der XML-Spezifikation unter www.w3.org.

Unterschiede zwischen der Web Publishing Engine und FileMaker Pro XML-Import/Export

Die Web Publishing Engine und FileMaker Pro gestatten Ihnen, XML-Daten mit FileMaker-Datenbanken zu verwenden. Es gibt jedoch einige wesentliche Unterschiede zwischen den beiden Methoden:

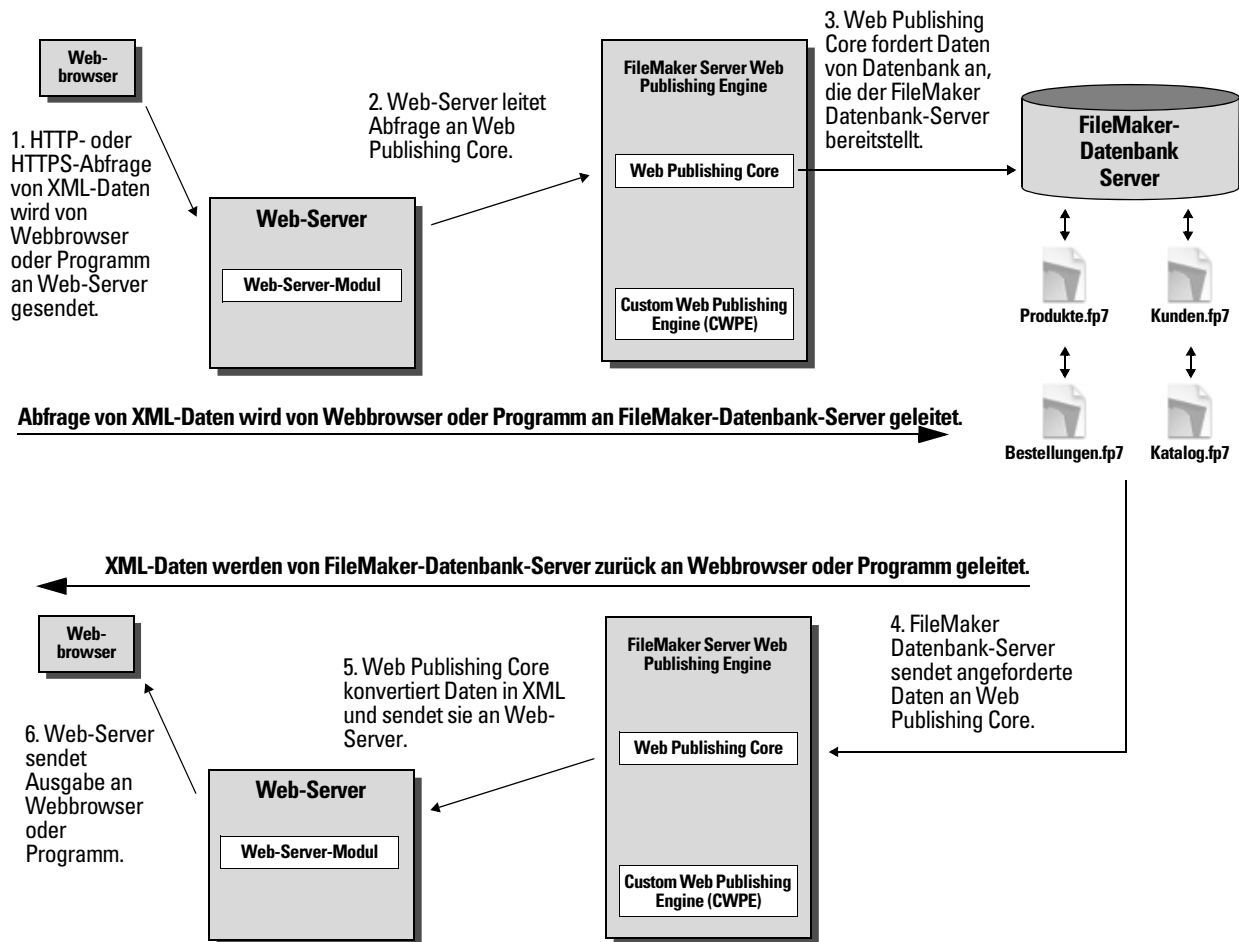
- Für den Zugriff auf XML-Daten und XSLT-Web Publishing unterstützt die Web Publishing Engine die `fmresultset`-, die `FMPXMLRESULT`- und die `FMPXMLLAYOUT`-Grammatik. Für XML-Import verwendet FileMaker Pro die `FMPXMLRESULT`-Grammatik und für den Export die `FMPXMLRESULT`- oder `FMPDSORESLT`-Grammatik. Weitere Informationen finden Sie unter „Zugriff auf XML-Daten über die Web Publishing Engine“ auf Seite 40.
- Um mit der Web Publishing Engine auf XML-Daten zuzugreifen, verwenden Sie einen Web Publishing Engine-Query-String in einem URL. Für den Import und Export von XML mit FileMaker Pro verwenden Sie FileMaker Pro-Menübefehle oder Scripts.

- Die Web Publishing Engine ist serverbasiert und kann auf demselben oder einem anderen Host als FileMaker Server installiert werden. FileMaker Pro XML-Import und -Export ist desktopbasiert.
- Sie können dynamisch auf XML-Daten von FileMaker-Datenbanken zugreifen, indem Sie URL-Abfragen mit der Web Publishing Engine verwenden. Die XML-Exportfunktion von FileMaker Pro erzeugt eine zuvor angegebene XML-Datendatei.
- Das Arbeiten mit XML-Daten über die Web Publishing Engine ist ein interaktiver Vorgang. FileMaker Pro XML-Import und -Export erfolgt im Stapelbetrieb.
- Die Web Publishing Engine kann auf XML-Daten von einem FileMaker-Ausschnitt zugreifen, FileMaker Pro nicht.
- Die Web Publishing Engine kann auf Daten in einem Medienfeld zugreifen, FileMaker Pro nicht.
- Die Web Publishing Engine bietet Echtzeitzugriff auf FileMaker-Daten über HTTP oder HTTPS, FileMaker Pro nicht.

Hinweis Informationen über das Verwenden von FileMaker Pro für Import und Export von Daten in XML-Format finden Sie in der FileMaker Pro Hilfe.

Wie die Web Publishing Engine aus einer Abfrage XML-Daten generiert

Nachdem eine Abfrage von XML-Daten an den Web-Server gesendet wurde, fragt die Web Publishing Engine die FileMaker-Datenbank ab und gibt die Daten als XML-Dokument zurück.



Allgemeines Verfahren beim Zugriff auf XML-Daten von der Web Publishing Engine

Im Folgenden erhalten Sie einen Überblick über das Verfahren beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten in einer FileMaker-Datenbank:

1. Stellen Sie in FileMaker Server Admin Console sicher, dass XML Publishing aktiviert ist. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
2. Öffnen Sie in FileMaker Pro jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank das erweiterte Zugriffsrecht fmxml für XML Custom Web Publishing aktiviert ist. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 21.

Um auf XML-Daten in einem Ausschnitt zuzugreifen, stellen Sie die Ansicht für das Datenbanklayout auf Als Formular anzeigen oder Als Liste anzeigen ein. Wenn ein Benutzer oder Script die Ansicht des Datenbanklayouts zu Als Tabelle anzeigen ändert, kann nur auf den ersten Bezugsdatensatz (erste Zeile im Ausschnitt) als XML-Daten zugegriffen werden.

Die XML-Daten werden in einer Reihenfolge ausgegeben, die der Reihenfolge entspricht, in der die Feldobjekte dem Layout hinzugefügt wurden. Wenn die XML-Datenreihenfolge der Reihenfolge entsprechen soll, in der die Felder auf dem Bildschirm angezeigt werden (von oben nach unten, von links nach rechts), dann wählen Sie alle Felder aus, gruppieren Sie sie und heben Sie dann die Gruppierung wieder auf. Dieser Vorgang setzt die Layoutreihenfolge wieder zurück auf die Bildschirmreihenfolge.

3. Senden Sie durch ein HTML-Formular, einen HREF-Link oder ein Script in Ihrem Programm oder Ihrer Web-Seite eine HTTP- oder HTTPS-Abfrage in Form eines URLs mit Angabe der FileMaker XML-Grammatik, einen Query-Befehl und einen oder mehrere FileMaker-Query-Parameter an die Web Publishing Engine. Der URL kann auch in einem Browser eingegeben werden.

Informationen über die Angabe des URLs finden Sie im nächsten Abschnitt, „Erläuterung der URL-Syntax für XML-Daten und Medienobjekte“. Informationen über Query-Befehle und -Parameter finden Sie unter „Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings“ auf Seite 50, und in Anhang A, „Gültige Namen in Query-Strings“.

4. Die Web Publishing Engine verwendet die im URL angegebene Grammatik, um XML-Daten mit den Ergebnissen Ihrer Abfrage zu erzeugen (z. B. eine Gruppe von Datensätzen aus der Datenbank), und gibt diese an Ihr Programm oder Ihren Webbrowser zurück.
5. Der Webbrowser, falls er über einen XML-Parser verfügt, zeigt die Daten an bzw. das Programm verwendet die Daten wie von Ihnen angegeben.

Wenn Sie ein Client-seitiges Stylesheet angegeben haben, weist der Parser des Webbrowsers auch die Stylesheet-Anweisungen zu. Weitere Informationen finden Sie unter „Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung“ auf Seite 53.

Erläuterung der URL-Syntax für XML-Daten und Medienobjekte

Dieser Abschnitt beschreibt die URL-Syntax beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten und Medienobjekte in FileMaker-Datenbanken. Die URL-Syntax für die Verwendung von XSLT-Stylesheets unterscheidet sich von XML. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 56 und „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen“ auf Seite 57.

Erläuterung der URL-Syntax für XML-Daten

Die URL-Syntax beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten in FileMaker-Datenbanken lautet:

```
<scheme>://<host>[:<port>]/fmi/xml/<xml_grammar>.xml[?<query string>]
```

Dabei gilt:

- <scheme> kann das HTTP- oder HTTPS-Protokoll sein.
- <host> ist die IP-Adresse oder der Domänenname des Hosts, auf dem der Web-Server installiert ist.
- <port> ist optional und gibt den Port an, den der Web-Server überwacht. Ohne Portangabe wird der Standard-Port des Protokolls (Port 80 für HTTP oder Port 443 für HTTPS) verwendet.
- <xml_grammar> ist der Name der FileMaker-XML-Grammatik. Mögliche Werte sind fmresultset.xml, FMPXMLRESULT.xml, FMPXMLLAYOUT.xml oder FMPDSORESLUT.xml. Weitere Informationen hierzu finden Sie unter „Verwenden der fmresultset-Grammatik“ auf Seite 41 und „Verwenden anderer FileMaker XML-Grammatiken“ auf Seite 45.
- <query string> ist eine Kombination aus einem Query-Befehl und einem oder mehreren Query-Parametern für FileMaker XML Publishing. (Der Befehl –dbnames verlangt keine Parameter.) Weitere Informationen finden Sie unter „Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings“ auf Seite 50, und Anhang A, „Gültige Namen in Query-Strings“.

Hinweis Die URL-Syntax einschließlich der Namen von Query-Befehl und -Parametern unterscheidet Groß- und Kleinschreibung, mit Ausnahme von Teilen des Query-Strings. Der URL ist vorwiegend in Kleinbuchstaben mit Ausnahme der drei Grammatiknamen in Großbuchstaben: FMPXMLRESULT, FMPXMLLAYOUT und FMPDSORESLUT. Informationen über die Regeln für Groß-/Kleinschreibung im Query-String finden Sie unter „Richtlinien für die Verwendung von Query-Befehlen und -Parametern“ auf Seite 94.

Zwei URL-Beispiele für den Zugriff auf XML-Daten über die Web Publishing Engine:

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=products&-lay=sales&-findall
```

```
http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=products&-lay=sales&-findall
```

Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen

In einem generierten XML-Dokument für eine XML-Lösung unterscheidet sich die Syntax für den Verweis auf ein Medienobjekt für Medienfelder, die das tatsächliche Objekt in der Datenbank speichern, und für Medienfelder, die einen Verweis auf das Objekt speichern.

- Wenn ein Medienfeld das tatsächliche Objekt in der Datenbank speichert, verwendet das <data>-Element des Medienfelds die folgende relative URL-Syntax für den Verweis auf das Objekt:

```
<data>/fmi/xml/cnt/data.<extension>?<query string></data>
```

Dabei gilt: <extension> ist die Dateinamenerweiterung zur Identifizierung des Objekttyps wie beispielsweise .jpg. Die Dateinamenerweiterung stellt den MIME-Typ ein, damit der Webbrowser die Mediendaten korrekt identifizieren kann. Informationen über <query string> finden Sie im vorherigen Abschnitt, „Erläuterung der URL-Syntax für XML-Daten“.

Beispiel:

```
<data>/fmi/xml/cnt/data.jpg?-db=products&-lay=sales&-field=product_image(1)&-recid=2</data>
```

Hinweis In der generierten XML für ein Medienfeld ist der Wert für den Query-Parameter -field ein vollständig qualifizierter Feldname. Die Zahl in Klammern gibt die Wiederholungsnummer für das Medienfeld an und wird für Wiederhol- und für Einzelfelder generiert. Weitere Informationen finden Sie unter „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 95.

Verwenden Sie zum Abrufen der Mediendaten aus der Datenbank die folgende Syntax:

```
<schema>://<host>[:<port>]/fmi/xml/cnt/data.<extension>?<query string>
```

Informationen über <schema>, <host> oder <port> finden Sie im vorherigen Abschnitt, „Erläuterung der URL-Syntax für XML-Daten“.

Beispiel:

```
http://www.company.com/fmi/xml/cnt/data.jpg?-db=products&-lay=sales&-field=product_image(1)&-recid=2
```

- Wenn ein Medienfeld einen Dateiverweis anstelle eines tatsächlichen Objekts speichert, enthält das <data>-Element des Medienfelds einen relativen Pfad, der auf das Objekt verweist. Beispiel:

```
<data>/images/logo.jpg</data>
```

Hinweis Das Medienobjekt, auf das verwiesen wird, muss im Ordner FileMaker Pro Web gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird, und dann in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software kopiert oder verschoben werden. Weitere Informationen finden Sie unter „Veröffentlichen des Inhalts von Medienfeldern im Web“ auf Seite 23.

- Für ein leeres Medienfeld ist auch das <data>-Element leer.

Hinweis Die Syntax für Medienobjekte mit XML unterscheidet sich von der Syntax für Medienobjekte mit XSLT. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen“ auf Seite 57.

Erläuterung von URL-Textkodierung

Die URLs für den Zugriff auf XML-Daten und Medienfelder müssen in UTF-8 (Unicode Transformation 8 Bit)-Format kodiert werden. Informationen finden Sie unter „Erläuterung von UTF-8-kodierten Daten“ auf Seite 50.

Um z. B. den Wert des Info-Felds in *Matinée* zu ändern, könnten Sie folgenden URL verwenden:

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=members&-lay=relationships&-recid=2
&info= matin%C3%A9&-edit
```

In diesem Beispiel-URL ist %C3%A9 die URL-kodierte UTF-8-Darstellung des Zeichens é.

Weitere Informationen zur URL-Textkodierung finden Sie in der URL-Spezifikation unter www.w3.org.

Zugriff auf XML-Daten über die Web Publishing Engine

Um über die Web Publishing Engine auf XML-Daten zuzugreifen, verwenden Sie einen URL, der den Namen der zu verwendenden FileMaker-Grammatik, einen FileMaker-Query-Befehl und einen oder mehrere FileMaker-Query-Parameter angibt. Die Web Publishing Engine generiert XML-Daten aus Ihrer Datenbank, die nach einem der folgenden Typen von XML-Grammatiken formatiert werden:

- **fmresultset:** Dies ist die empfohlene Grammatik für die Web Publishing Engine. Sie ist flexibel und für die Erstellung von XSLT-Stylesheets mit bequemem Feldzugriff nach Name und einfacherer Änderung von relatedset- (Ausschnitt-) Daten optimiert. Diese Grammatik ist auch direkter mit FileMaker-Terminologie verbunden und bietet z. B. globale Speicheroptionen und Identifikation von Statistik- und Formelfeldern. Sie können diese Grammatik für den Zugriff auf XML-Daten und für XSLT-Stylesheets verwenden. Für einfacheres Web Publishing wurde diese Grammatik knapper formuliert als die FMPXMLRESULT-Grammatik. Weitere Informationen finden Sie unter „Verwenden der fmresultset-Grammatik“ auf Seite 41.
- **FMPXMLRESULT und FMPXMLLAYOUT:** Sie können auch die FMPXMLRESULT- und FMPXMLLAYOUT-Grammatiken mit der Web Publishing Engine für den Zugriff auf XML-Daten und für XSLT-Stylesheets verwenden. Um ein einziges Stylesheet sowohl für XML-Export als auch für Custom Web Publishing einzusetzen, müssen Sie die FMPXMLRESULT-Grammatik benutzen. Für den Zugriff auf Wertelisten und Feldanzeigedaten in Layouts müssen Sie die FMPXMLLAYOUT-Grammatik verwenden. Weitere Informationen finden Sie unter „Verwenden anderer FileMaker XML-Grammatiken“ auf Seite 45.
- **FMPDSORESET:** Die FMPDSORESET-Grammatik, die in FileMaker Pro für den Export von XML unterstützt wird, steht für den Zugriff auf XML-Daten über die Web Publishing Engine nicht zur Verfügung. Die FMPDSORESET-Grammatik wird für XSLT-Stylesheets nicht unterstützt. Informationen über die FMPDSORESET-Grammatik finden Sie in der FileMaker Pro Hilfe.

Abhängig von der Grammatik, die Sie in der URL-Abfrage angegeben haben, generiert die Web Publishing Engine mit einer der folgenden Grammatiken ein XML-Dokument. Jedes XML-Dokument enthält eine XML-Standard-Namespace-Deklaration für die Grammatik. Informationen finden Sie im nächsten Abschnitt, „Erläuterung von Namensräumen für FileMaker XML“. Verwenden Sie eine dieser Grammatiken in Ihrem Dokument oder auf Ihrer Web-Seite, um FileMaker-Daten im XML-Format anzuzeigen und zu bearbeiten.

Hinweis Von der Web Publishing Engine generierte XML-Daten werden im UTF-8-Format (Unicode Transformation Format 8) kodiert. Weitere Informationen finden Sie unter „Erläuterung von UTF-8-kodierten Daten“ auf Seite 50.

Erläuterung von Namensräumen für FileMaker XML

Mithilfe eindeutiger XML-Namensräume können Sie XML-Tags der richtigen Anwendung zuordnen. Wenn Ihr XML-Dokument beispielsweise zwei <DATABASE>-Elemente enthält, eines für FileMaker Pro-XML-Daten, das andere für Oracle-XML-Daten, kann anhand der Namensräume festgestellt werden, welches <DATABASE>-Element zu welchen XML-Daten gehört.

Die Web Publishing Engine generiert für jede Grammatik einen Standard-Namespace.

Grammatik	Generierter Standard-Namespace
fmresultset	xmlns="http://www.filemaker.com/xml/fmresultset"
FMPXMLRESULT	xmlns="http://www.filemaker.com/ fmpxmlresult"
FMPXMLLAYOUT	xmlns="http://www.filemaker.com/fmpxmllayout"

Erläuterung der FileMaker-Datenbankfehlercodes

Die Web Publishing Engine gibt in den „error code“-Elementen am Beginn jedes XML-Dokuments einen Fehlercode für einen etwaigen Fehler zurück, der im zuletzt ausgeführten Query-Befehl aufgetreten ist. Null (0) bedeutet keinen Fehler.

Grammatik	Verwendete Syntax
fmresultset	<error code="0"></error>
FMPXMLRESULT	<ERRORCODE>0</ERRORCODE>
FMPDSORESLT	<ERRORCODE>0</ERRORCODE>

Das „error code“-Element im XML-Dokument gibt Fehler an, die die Datenbank und Query-Strings betreffen. Andere Fehlertypen können für XSLT-Stylesheets auftreten und werden anders behandelt. Weitere Informationen finden Sie unter Anhang B, „Fehlercodes für Custom Web Publishing“.

Abrufen der Document Type Definitions für die FileMaker-Grammatiken

Sie können die Document Type Definitions (DTDs) für die FileMaker-Grammatiken mithilfe einer HTTP-Abfrage abrufen.

Grammatik	HTTP-Abfrage
fmresultset	http://<host>[:<port>]/fmi/xml/fmresultset.dtd
FMPXMLRESULT	http://<host>[:<port>]/fmi/xml/FMPXMLRESULT.dtd
FMPXMLLAYOUT	http://<Host>[:<Port>]/fmi/xml/FMPXMLLAYOUT.dtd
FMPDSORESLT	http://<host>[:<port>]/fmi/xml/FMPDSORESLT.dtd?-db=<database>&-lay=<layout>

Verwenden der fmresultset-Grammatik

Die XML-Elementnamen in dieser Grammatik verwenden FileMaker-Terminologie und die Speicherung von Feldern erfolgt getrennt von den Feldtypen. Die Grammatik umfasst auch die Fähigkeit, Statistik-, Formel- und Variablenfelder zu identifizieren.

Geben Sie zur Verwendung der fmresultset-Grammatik den folgenden Namen der fmresultset-Grammatik im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

fmresultset.xml

Beispiel:

http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family&-findall

Hinweis Geben Sie die fmresultset-Grammatik unbedingt in Kleinbuchstaben an.

Die Web Publishing Engine generiert ein XML-Dokument mithilfe der fmresultset-Grammatik. Im XML-Dokument verweist die Web Publishing Engine auf die Document Type Definition für die fmresultset-Grammatik in der Anweisung <!DOCTYPE> in der zweiten Zeile des Dokuments, unmittelbar nach der Anweisung <?xml...?>. Die Anweisung <!DOCTYPE> gibt den URL für das Herunterladen der DTD für die fmresultset-Grammatik an.

Beschreibung der Elemente der fmresultset-Grammatik

Die fmresultset-Grammatik besteht hauptsächlich aus dem Element <datasource>, dem Element <metadata> und dem Element <resultset>.

Element <datasource>

In der fmresultset-Grammatik enthält das <datasource>-Element die Attribute „table“, „layout“, „date-format“, „time-format“, „timestamp-format“, „total-count“ und „database“.

- Das Attribut „date-format“ des <datasource>-Elements gibt das Format von Datumsangaben im XML-Dokument an:

MM.tt.jjjj

Dabei gilt:

- MM ist der zweistellige Wert für den Monat (01 bis 12, wobei 01 für Januar und 12 für Dezember steht).
- tt ist der zweistellige Wert für den Tag des Monats (00 bis 31).
- jjjj ist der vierstellige Wert für das Jahr.
- Das Attribut „time-format“ des <datasource>-Elements gibt das Format von Zeitangaben im XML-Dokument an.

HH:mm:ss

Dabei gilt:

- HH ist der zweistellige Wert für Stunden (00 bis 23 im 24-Stunden-Format).
- mm ist der zweistellige Wert für Minuten (00 bis 59).
- ss ist der zweistellige Wert für Sekunden (00 bis 59).
- Das Attribut „timestamp-format“ des <datasource>-Elements gibt das Format von Datums- und Zeitangaben in einem Zeitstempel an.

MM.tt.jjjj HH:mm:ss

Element <metadata>

Das <metadata>-Element der fmresultset-Grammatik enthält ein oder mehrere <field-definition>- und <relatedset-definition>-Elemente, jeweils mit Attributen für eines der Felder in der Ergebnismenge.

Die Attribute <field-definition> geben an:

- ob das Feld ein Feld mit automatischer Eingabe ist („yes“ oder „no“)
- ob das Feld ein Feld mit vierstelligem Jahr ist („yes“ oder „no“)
- ob das Feld ein globales Feld ist („yes“ oder „no“)
- die maximale Anzahl an Wiederholwerten (Attribut max-repeat)
- die maximale Anzahl an zulässigen Zeichen (Attribut max-characters)
- ob das Feld ein nicht leeres Feld ist („yes“ oder „no“)
- ob das Feld ein Feld nur mit numerischen Daten ist („yes“ oder „no“)
- result („text“, „number“, „date“, „time“, „timestamp“ oder „container“)
- ob das Feld ein Uhrzeit-des-Tages-Feld ist („yes“ oder „no“)

- type („normal“, „calculation“ oder „summary“)
- den Feldnamen (bei Bedarf vollständig qualifiziert)

Das <relatedset-definition>-Element repräsentiert einen Ausschnitt. Jedes Bezugsfeld in einem Ausschnitt wird durch das <field-definition>-Element innerhalb des <relatedset-definition>-Elements dargestellt. Wenn ein Ausschnitt mehrere Bezugsfelder enthält, werden die Felddefinitionen für die Bezugsfelder in einem einzigen <relatedset-definition>-Element gruppiert.

Element <resultset>

Das <resultset>-Element enthält alle <record>-Elemente, die als Ergebnis einer Query zurückgegeben wurden, sowie ein Attribut für die Gesamtzahl der gefundenen Datensätze. Jedes <record>-Element enthält die Felddaten für einen Datensatz in der Ergebnismenge – einschließlich der Attribute mod-id und record-id für den Datensatz – sowie das <data>-Element mit den Daten für ein Feld im Datensatz.

Jeder Datensatz in einem Ausschnitt wird durch ein <record>-Element innerhalb des <relatedset>-Elements dargestellt. Das Attribut „count“ des <relatedset>-Elements gibt die Anzahl der Datensätze in einem Ausschnitt an und das Attribut „table“ gibt die mit dem Ausschnitt verbundene Tabelle an.

Beispiel für XML-Daten in der fmresultset-Grammatik

Nachfolgend erhalten Sie ein Beispiel für XML-Daten, die mit der fmresultset-Grammatik generiert wurden.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fmresultset PUBLIC "-//FMI//DTD fmresultset//EN" ""http://localhost:16014/fmi/xml/fmresultset.dtd">
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset" version="1.0">
  <error code="0" />
  <product build="31.12.2012" name="FileMaker Web Publishing Engine" version="0.0.0.0" />
  <datasource database="art" date-format="MM/dd/yyyy" layout="web3" table="art" time-format="HH:mm:ss" timestamp-format="MM/dd/yyyy HH:mm:ss" total-count="12" />
  <metadata>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="Title" not-empty="no" numeric-only="no" result="text" time-of-day="no" type="normal" />
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="Artist" not-empty="no" numeric-only="no" result="text" time-of-day="no" type="normal" />
    <relatedset-definition table="artlocations">
      <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="artlocations::Location" not-empty="no" numeric-only="no" result="text" time-of-day="no" type="normal" />
      <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="artlocations::Date" not-empty="no" numeric-only="no" result="date" time-of-day="no" type="normal" />
    </relatedset-definition>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="Style" not-empty="no" numeric-only="no" result="text" time-of-day="no" type="normal" />
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-repeat="1" name="Length" not-empty="no" numeric-only="no" result="number" time-of-day="no" type="calculation" />
  </metadata>
  <resultset count="1" fetch-size="1">
    <record mod-id="6" record-id="14">
      <field name="Title">
        <data>Spring in Giverny 3</data>
      </field>
      <field name="Artist">
        <data>Claude Monet</data>
      </field>
      <relatedset count="0" table="artlocations" />
      <field name="Style">
        <data />
      </field>
      <field name="length">
        <data>19</data>
      </field>
    </record>
  </resultset>
</fmresultset>
```

Verwenden anderer FileMaker XML-Grammatiken

Die anderen FileMaker XML-Grammatiken enthalten Informationen über Feldtypen, Wertelisten und Layouts. FMPXMLRESULT und fmresultset sind hinsichtlich der Funktionalität äquivalent. Für den Zugriff auf Wertelisten und Feldanzeigedaten in Layouts müssen Sie die FMPXMMLAYOUT-Grammatik verwenden. Die FMPXMLRESULT- und FMPXMMLAYOUT-Grammatiken sind kompakter für den Datenaustausch.

Geben Sie zur Verwendung der FMPXMLRESULT-Grammatik den folgenden Grammatiknamen im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

FMPXMLRESULT.xml

Beispiel:

<http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=employees&-lay=family&-findall>

Geben Sie zur Verwendung der FMPXMMLAYOUT-Grammatik den folgenden Grammatiknamen mit dem Query-Befehl -view im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

FMPXMMLAYOUT.xml

Beispiel:

<http://192.168.123.101/fmi/xml/FMPXMMLAYOUT.xml?-db=employees&-lay=family&-view>

Hinweis Geben Sie die FMPXMLRESULT- und die FMPXMMLAYOUT-Grammatik unbedingt in Großbuchstaben an.

Im generierten XML-Dokument verweist die Web Publishing Engine auf die Document Type Definition für die fmresultset-Grammatik in der Anweisung <!DOCTYPE> in der zweiten Zeile des Dokuments, unmittelbar nach der Anweisung <?xml...?>. Die Anweisung <!DOCTYPE> gibt den URL für das Herunterladen der DTD für die Grammatik an.

Beschreibung der Elemente in der FMPXMLRESULT-Grammatik

In der FMPXMLRESULT-Grammatik enthält das <DATABASE>-Element die Attribute NAME, RECORDS, DATEFORMAT, LAYOUT und TIMEFORMAT.

Das Attribut DATEFORMAT des <DATABASE>-Elements gibt das Datumsformat im XML-Dokument an. Das Attribut TIMEFORMAT des <DATABASE>-Elements gibt das Zeitformat im XML-Dokument an. Datums- und Zeitformate der Grammatiken FMPXMLRESULT und fmresultset sind gleich (siehe dazu die Tabellen unter „Beschreibung der Elemente der fmresultset-Grammatik“ auf Seite 42).

Das <METADATA>-Element der FMPXMLRESULT-Grammatik enthält ein oder mehrere <FIELD>-Elemente. Jedes FIELD-Element enthält Daten für ein Feld bzw. eine Spalte in der Ergebnismenge einschließlich des Feldnamens, wie er in der Datenbank definiert ist, den Feldtyp, die Ja/Nein-Zulässigkeit für leere Felder (Attribut EMPTYOK) und die maximale Anzahl von Wiederholwerten (Attribut MAXREPEAT). Gültige Werte für Feldtypen sind TEXT, NUMBER, DATE, TIME, TIMESTAMP und CONTAINER.

Das <RESULTSET>-Element enthält alle <ROW>-Elemente, die als Ergebnis einer Query zurückgegeben wurden, sowie ein Attribut für die Gesamtzahl der gefundenen Datensätze. Jedes <ROW>-Element enthält die Feld-/Spalten-Daten für jede Zeile in der Ergebnismenge. Diese Daten umfassen RECORDID und MODID für die Zeile (siehe „Query-Parameter -modid (Änderungs-ID)“ auf Seite 105) und das <COL>-Element. Das <COL>-Element enthält die Daten für ein Feld/eine Spalte in der Zeile, wobei mehrere <DATA>-Elemente einen der Werte in einem Wiederhol- oder Ausschnittfeld repräsentieren.

Beispiel für XML-Daten in der FMPXMLRESULT-Grammatik

Nachfolgend sehen Sie ein Beispiel für XML-Daten, die mit der FMPXMLRESULT-Grammatik generiert wurden.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMLRESULT PUBLIC "-//FMI//DTD FMPXMLRESULT//EN" ""http://localhost:16014/fmi/xml/
FMPXMLRESULT.dtd">
<FMPXMLRESULT xmlns="http://www.filemaker.com/fmpxmlresult">
  <ERRORCODE>0</ERRORCODE>
  <PRODUCT BUILD="31.12.2012" NAME="FileMaker Web Publishing Engine" VERSION="0.0.0.0" />
  <DATABASE DATEFORMAT="MM/dd/yyyy" LAYOUT="web" NAME="art" RECORDS="12" TIMEFORMAT="HH:mm:ss" />
  <METADATA>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Title" TYPE="TEXT" />
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Artist" TYPE="TEXT" />
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Image" TYPE="CONTAINER" />
  </METADATA>
  <RESULTSET FOUND="1">
    <ROW MODID="6" RECORDID="15">
      <COL>
        <DATA>Spring in Giverny 4</DATA>
      </COL>
      <COL>
        <DATA>Claude Monet</DATA>
      </COL>
      <COL>
        <DATA>/fmi/xml/cnt/data.jpg?-db=art&-lay=web&-recid=15&-field=Image(1)</DATA>
      </COL>
    </ROW>
  </RESULTSET>
</FMPXMLRESULT>
```

Die Reihenfolge der <COL>-Elemente entspricht der Reihenfolge der <FIELD>-Elemente im <METADATA>-Element – z. B. entspricht die Reihenfolge der Felder „Title“ und „Artist“ im <METADATA>-Element der Reihenfolge von „Village Market“ und „Camille Pissarro“ in den Elementen <RESULTSET> und <ROW>.

Beschreibung der Elemente der Grammatik FMPXMLLAYOUT

In der FMPXMLLAYOUT-Grammatik enthält das <LAYOUT>-Element den Namen des Layouts, den Namen der Datenbank sowie <FIELD>-Elemente für jedes Feld im entsprechenden Layout der Datenbank. Jedes <FIELD>-Element beschreibt den Feldtyp und enthält das Attribut VALUELIST für jede Werteliste des Feldes.

Das <VALUELISTS>-Element enthält ein oder mehrere <VALUELIST>-Elemente für jede Werteliste im Layout – jedes VALUELIST-Element enthält den Namen der Werteliste und ein <VALUE>-Element für jeden Wert in der Liste.

Abhängig von den im Dialogfeld Felder für Werteliste angeben in der FileMaker-Datenbank gewählten Optionen enthält das Element <VALUE> ein DISPLAY-Attribut, das den Wert nur im ersten Feld, im zweiten Feld oder in beiden Feldern einer Werteliste enthält. Nehmen Sie zum Beispiel an, dass das erste Feld in einer Werteliste die ID-Nummer des Kunststils (wie z. B. „100“) und das zweite Feld den zugehörigen Namen des Kunststils (wie z. B. „Impressionismus“) enthält. Hier finden Sie eine Zusammenfassung des Inhalts des Attributs DISPLAY, wenn die verschiedenen Kombinationen der Optionen im Dialogfeld Felder für Werteliste angeben ausgewählt sind:

- Wenn Auch Werte aus zweitem Feld anzeigen nicht ausgewählt ist, enthält das Attribut DISPLAY nur den Wert im ersten Feld einer Werteliste. Im folgenden XML-Datenbeispiel enthält das Attribut DISPLAY nur die ID-Nummer des Kunststils:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="100">100</VALUE>
    <VALUE DISPLAY="101">101</VALUE>
    <VALUE DISPLAY="102">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

- Wenn sowohl Auch Werte aus zweitem Feld anzeigen als auch Werte nur aus dem zweiten Feld anzeigen ausgewählt ist, enthält das DISPLAY-Attribut nur den Wert im zweiten Feld. Im folgenden XML-Datenbeispiel enthält das Attribut DISPLAY nur den Namen des Kunststils:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="Impressionism">100</VALUE>
    <VALUE DISPLAY="Cubism">101</VALUE>
    <VALUE DISPLAY="Abstract">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

- Wenn Auch Werte aus zweitem Feld anzeigen ausgewählt ist und Werte nur aus dem zweiten Feld anzeigen nicht ausgewählt ist, enthält das DISPLAY-Attribut die Werte in beiden Feldern einer Werteliste. Im folgenden XML-Datenbeispiel enthält das Attribut DISPLAY sowohl die ID-Nummer als auch den Namen des Kunststils:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="100 Impressionism">100</VALUE>
    <VALUE DISPLAY="101 Cubism">101</VALUE>
    <VALUE DISPLAY="102 Abstract">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

Für Datums-, Zeit- und Zeitstempelfelder werden Daten für Wertelisten über das „fm“-Format für diesen Feldtyp formatiert. Die „fm“-Formate sind MM/tt/jjjj für Datum, HH:mm:ss für Zeit und MM/tt/jjjj HH:mm:ss für Zeitstempel. Weitere Informationen finden Sie unter „Verwenden der Datums-, Zeit- und Tages-Erweiterungsfunktionen“ auf Seite 78. Wenn zum Beispiel eine Werteliste „Geburtstage“ für ein Einblendmenü für ein Feld „Geburtstag“ eines Layouts verwendet wird und das Feld „Geburtstag“ ein Datumsfeld ist, werden alle für diese Werteliste ausgegebenen Werte im Datumsformat „fm“ ausgegeben.

Hinweis Wenn zwei Felder mit unterschiedlichem Feldtyp in einem Layout die gleiche Werteliste verwenden, bestimmt das erste Feld das Format der Daten der Werteliste.

XML-Daten in der FMPXMMLAYOUT-Grammatik – Beispiel

Im Folgenden finden Sie ein Beispiel für XML-Daten, die mit der FMPXMMLAYOUT-Grammatik generiert wurden.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMMLAYOUT PUBLIC "-//FMI//DTD FMPXMMLAYOUT//EN" ""http://localhost:16014/fmi/xml/
FMPXMMLAYOUT.dtd">
<FMPXMMLAYOUT xmlns="http://www.filemaker.com/fmpxmmlayout">
<ERRORCODE>0</ERRORCODE>
<PRODUCT BUILD="31.12.2012" NAME="FileMaker Web Publishing Engine" VERSION="0.0.0.0" />
<LAYOUT DATABASE="art" NAME="web2">
<FIELD NAME="Title">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
</FIELD>
<FIELD NAME="Artist">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
</FIELD>
<FIELD NAME="Image">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
</FIELD>
<FIELD NAME="artlocations::Location">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
</FIELD>
<FIELD NAME="artlocations::Date">
  <STYLE TYPE="EDITTEXT" VALUELIST="" />
</FIELD>
<FIELD NAME="Style">
  <STYLE TYPE="POPUPMENU" VALUELIST="style"/>
</FIELD>
</LAYOUT>
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="Impressionism">100</VALUE>
    <VALUE DISPLAY="Cubism">101</VALUE>
    <VALUE DISPLAY="Abstract">102</VALUE>
  </VALUELIST>
</VALUELISTS>
</FMPXMMLAYOUT>
```

Erläuterung von UTF-8-kodierten Daten

Alle von der Web Publishing Engine generierten XML-Daten werden im UTF-8-Format (Unicode Transformation Format 8) kodiert. Dieses Format komprimiert ASCII-Zeichen im Unicode-Standardformat von 16 Bit auf 8 Bit. Für Unicode und UTF-8 sind XML-Parser erforderlich.

UTF-8-Kodierung unterstützt die direkte Darstellung der Werte 0-127 für den ASCII-Standardzeichensatz in der englischen Sprache und bietet Multibyte-Kodierungen für Unicode-Zeichen mit höheren Werten.

Hinweis Stellen Sie sicher, dass Sie einen Webbrowser oder ein Textverarbeitungsprogramm verwenden, das UTF-8-Dateien unterstützt.

Das Kodierungsformat UTF-8 hat die folgenden Eigenschaften:

- Alle ASCII-Zeichen sind UTF-8-Zeichen von der Größe eines Byte. Eine gültige ASCII-Zeichenkette ist eine gültige UTF-8-Zeichenkette.
- Jedes Nicht-ASCII-Zeichen (beliebiges Zeichen mit gesetztem höherwertigem Bit) ist Teil eines Multibyte-Zeichens.
- Das erste Byte eines UTF-8-Zeichens gibt die Anzahl der zusätzlichen Bytes im Zeichen an.
- Das erste Byte eines Multibyte-Zeichens lässt sich leicht vom darauf folgenden Byte unterscheiden, wodurch sich der Beginn eines Zeichens von einer beliebigen Position in einem Datenstrom einfach ermitteln lässt.
- Eine Konvertierung zwischen UTF-8 und Unicode ist einfach vorzunehmen.
- Die UTF-8-Kodierung ist relativ kompakt. Bei Text, der einen hohen Prozentsatz an ASCII-Zeichen enthält, ist sie kompakter als Unicode. Im schlimmsten Fall ist ein UTF-8-String nur 50 % größer als der entsprechende Unicode-String.

Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings

Um XML-Daten aus einer FileMaker-Datenbank abzufragen, verwenden Sie die FileMaker-Query-Befehle und -Parameter in einem Query-String. Sie können beispielsweise den Query-Befehl `-findall` im folgenden Query-String in einem URL verwenden, um eine Liste aller Produkte aus einer FileMaker-Datenbank mit dem Namen `products` anzufordern:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=products-lay=sales&-findall
```

Ein Query-String darf nur einen Query-Befehl, z. B. `-new`, enthalten. Die meisten Query-Befehle verlangen auch mehrere entsprechende Query-Parameter im Query-String. Beispielsweise verlangen alle Query-Befehle außer `-dbnames` den Parameter `-db`, der die abzufragende Datenbank angibt.

Sie können auch Query-Befehle und -Parameter in einem URL oder in einer `<?xslt-cwp-query?>`-Verarbeitungsanweisung in einem FileMaker XSLT-Stylesheet verwenden. Weitere Informationen finden Sie unter Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Dieser Abschnitt bietet eine Zusammenfassung der FileMaker Query-Befehle und -Parameter. Weitere Informationen über ihre Verwendung in einem Query-String finden Sie in Anhang A, „Gültige Namen in Query-Strings“.

Hinweis Die Web Publishing Engine unterstützt auch einen zusätzlichen Query-Befehl (`-process`) sowie drei Query-Parameter, die nur zur Verwendung mit FileMaker XSLT-Stylesheets definiert sind. Weitere Informationen finden Sie unter „Verwenden von Query-Strings in FileMaker XSLT-Stylesheets“ auf Seite 58.

Verwenden Sie diesen Query-Befehlsnamen

Zur Ausführung dieses Befehls

<code>-dbnames</code>	Namen aller bereitgestellten und im Web gemeinsam genutzten Datenbanken abrufen.
<code>-delete</code>	Datensatz löschen.
<code>-dup</code>	Datensatz duplizieren.
<code>-edit</code>	Datensatz bearbeiten.
<code>-find</code>	Datensatz/Datensätze suchen.
<code>-findall</code>	Nach allen Datensätzen suchen.
<code>-findany</code>	Zufällig ermittelten Datensatz anzeigen.
<code>-findquery</code>	Komplexe oder zusammengesetzte Suchabfrage durchführen.
<code>-layoutnames</code>	Namen aller verfügbaren Layouts für eine bereitgestellte und im Web gemeinsam genutzte Datenbank abrufen.
<code>-new</code>	Neuen Datensatz hinzufügen.
<code>-scriptnames</code>	Namen aller verfügbaren Scripts für eine bereitgestellte und im Web gemeinsam genutzte Datenbank abrufen.
<code>-view</code>	Layout-Informationen aus einer Datenbank abrufen, wenn die FMPXMLLAYOUT-Grammatik angegeben ist. Ruft den Abschnitt <code><metadata></code> des XML-Dokuments und eine leere Datensatzmenge ab, wenn die <code>fmresultset</code> - oder <code>FMPXMLRESULT</code> -Grammatik angegeben ist.

Verwenden Sie diese Query-Parameternamen

Mit diesen Query-Befehlen

<code>-db</code> (Datenbankname)	Erforderlich für alle Query-Befehle mit Ausnahme von <code>-dbnames</code> und <code>-process</code> (nur XSLT-Abfragen)
<code>-delete.related</code>	Optional bei <code>-edit</code>
<code>-field</code>	Erforderlich zur Angabe eines Felds in einem URL für Medienabfragen. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen“ auf Seite 38
Feldname	Mindestens ein Feldname ist für <code>-edit</code> erforderlich. Optional bei <code>-find</code> . Weitere Informationen finden Sie unter „Query-Parameter „fieldname“ (Nicht-Medienfeldname)“ auf Seite 102
Feldname.op (Operator)	Optional bei <code>-find</code>
<code>-lay</code> (Layoutname)	Erforderlich für alle Query-Befehle mit Ausnahme von <code>-dbnames</code> , <code>-layoutnames</code> , <code>-scriptnames</code> und <code>-process</code> (nur XSLT-Abfragen)
<code>-lay.response</code> (Layout für XML-Antwort wechseln)	Optional für alle Query-Befehle mit Ausnahme von <code>-dbnames</code> , <code>-layoutnames</code> , <code>-scriptnames</code> und <code>-process</code> (nur XSLT-Abfragen)
<code>-lop</code> (logischer Operator)	Optional bei <code>-find</code>
<code>-max</code> (maximale Datensätze)	Optional bei <code>-find</code> , <code>-findall</code>

Verwenden Sie diese Query-Parameternamen	Mit diesen Query-Befehlen
-modid (Änderungs-ID)	Optional bei -edit
-query	Erforderlich bei zusammengesetzten Suchabfragen -findquery.
-recid (Datensatz-ID)	Erforderlich bei -edit, -delete, -dup. Optional bei -find
-relatedsets.filter	Optional bei -find, -edit, -new, -dup und -findquery.
-relatedsets.max	Optional bei -find, -edit, -new, -dup und -findquery.
-script (Script ausführen)	Optional bei -find, -findall, -findany, -new, -edit, -delete, -dup, -view
-script.param (einen Parameterwert an das durch -script angegebene Script übergeben)	Optional bei -script
-script.prefind (Script vor -find, -findany und -findall ausführen)	Optional bei -find, -findany, -findall
-script.prefind.param (einen Parameterwert an das durch -script.prefind angegebene Script übergeben)	Optional bei -script.prefind
-script.presort (Script vor Sortierung ausführen)	Optional bei -find, -findall
-script.presort.param (einen Parameterwert an das durch -script.presort angegebene Script übergeben)	Optional bei -script.presort
-skip (Datensätze überspringen)	Optional bei -find, -findall
-sortfield.[1-9] (Sortierfeld)	Optional bei -find, -findall
-sortorder.[1-9] (Sortierfolge)	Optional bei -find, -findall
-stylehref (Stylesheet HREF)	Optional für alle Query-Befehle (zur Angabe eines Stylesheet-URLs für -styletype)
-styletype (Stylesheet-Typ)	Optional für alle Query-Befehle (zur Angabe eines Client-seitigen Stylesheets)

Wechseln des Layouts für eine XML-Antwort

Der Query-Parameter -lay gibt das Layout an, das Sie beim Abfragen von XML-Daten verwenden wollen. Häufig eignet sich dasselbe Layout für die Verarbeitung der Daten, die von der Abfrage geliefert werden. In einigen Fällen sollten Sie die Daten in einem Layout durchsuchen, das Felder enthält, die aus Sicherheitsgründen in keinem Layout vorhanden sind, mit dem die Ergebnisse angezeigt werden. (Um Daten in einem Feld zu suchen, muss das Feld auf das Layout gesetzt werden, das Sie in der XML-Abfrage angeben.)

Um für die Anzeige der XML-Antwort ein anderes Layout als für die Verarbeitung der XML-Abfrage zu verwenden, können Sie den optionalen Query-Parameter -lay.response angeben.

Die folgende Abfrage sucht z. B. Werte größer 100,000 im Feld Salary im Layout Budget. Die gefundenen Daten werden im Layout ExecList angezeigt, das das Feld Salary nicht enthält.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=Budget&Salary=100000&Salary.op=gt&-find
&-lay.response=VorstListe
```

Erläuterung der Verarbeitung einer XML-Abfrage

Es gibt mehrere Query-Parameter, die die Verarbeitung einer XML-Abfrage und die Generierung eines XML-Dokuments beeinflussen.

FileMaker Server und die Web Publishing Engine verarbeiten eine XML-Abfrage in dieser Reihenfolge:

1. Verarbeitung des `-lay` -Query-Parameters
2. Festlegen der globalen Feldwerte, die in der Query angegeben sind (der Teil `.global=` des URLs)
3. Verarbeitung des Query-Parameters `-script.prefind`, falls angegeben
4. Verarbeitung der Query-Befehle wie z. B. `-find` oder `-new`
5. Verarbeitung des Query-Parameters `-script.presort`, falls angegeben
6. Sortieren der Ergebnisdaten, falls eine Sortierung angegeben wurde
7. Verarbeitung des Query-Parameters `-lay.response`, um das Layout zu wechseln, falls angegeben
8. Verarbeitung des Query-Parameters `-script`, falls angegeben
9. Generieren des XML-Dokuments.

Wenn einer der obigen Schritte einen Fehlercode erzeugt, stoppt die Abfrageverarbeitung und etwaige nachfolgende Schritte werden nicht ausgeführt. Jedoch bleiben alle vorangehenden Schritte in der Abfrage ausgeführt.

Betrachten Sie z. B. eine Abfrage, die den aktuellen Datensatz löscht, die Datensätze sortiert und dann ein Script ausführt. Wenn der Parameter `-sortfield` ein nicht vorhandenes Feld angibt, löscht die Abfrage den aktuellen Datensatz und gibt dann Fehlercode 102 (Feld fehlt) zurück, führt aber das Script nicht aus.

Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung

Die Web Publishing Engine unterstützt Server-seitige Verarbeitung eines XSLT-Stylesheets und gestattet Ihnen auch die Verwendung eines Query-Parameters, der Client-seitige Stylesheet-Verarbeitung angibt.

Es ist wichtig, die Unterschiede zwischen den beiden Arten der Stylesheet-Verarbeitung sowie die Implikationen für die Sicherheit bei Client-seitiger Verarbeitung zu verstehen. Server-seitige Verarbeitung gibt Web-Benutzern keinen Zugang zu den ungefilterten XML-Daten und ist daher sicherer als Client-seitige Verarbeitung. Bei Server-seitiger Verarbeitung werden die Daten in einer Form präsentiert, die der Eigentümer der Daten oder der Verfasser des XSLT-Stylesheets für angemessen hält. Server-seitige Verarbeitung verbirgt die Datenbanknamen, Feldnamen und andere Implementierungsdetails vor Web-Benutzern. Server-seitige Verarbeitung kann auch für die Angabe statisch definierter Query-Parameter benutzt werden, die die Verwendung nicht zugelassener Query-Befehle und Query-Parameter verhindert, z. B. Datenbanknamen. Weitere Informationen finden Sie unter Kapitel 4, „Einführung in Custom Web Publishing mit XSLT“ und Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Wenn Ihre Lösung Client-seitige Stylesheet-Verarbeitung erfordert, können Sie über die Web Publishing Engine eine Anweisung für die XML-Stylesheet-Verarbeitung mit jeder Grammatik generieren, indem Sie die Parameter `-styletype` und `-stylehref` in die FileMaker-Query-Stringabfrage einfügen. Sie können Cascading Style Sheets (CSS) oder XSLT-Stylesheets für die Anzeige Ihres XML-Dokuments verwenden.

- Mit dem Parameter `–styletype` wird der Wert des `type`-Attributs festgelegt (`type=text/css` oder `type=text/xsl`).
- Mit dem Parameter `–stylehref` wird der Wert des `HREF`-Attributs festgelegt, das die Position des Stylesheets mit einem absoluten Pfad angibt. Beispiel: `href=/mystylesheet.css` oder `href=/stylesheets/mystylesheet.xml`
Das Stylesheet kann einen beliebigen Namen haben, aber die Erweiterung muss `.css` oder `.xsl` sein.

Beispiel für einen FileMaker-Query-String, der Client-seitige Stylesheet-Verarbeitung erzeugt:

```
http://localhost/fmi/xml/fmresultset.xml?-db=products-lay=sales&-findall&-styletype=text/xsl&-stylehref=/mystylesheet.xml
```

Hinweis Der Schrägstrich / in `–stylehref=/document.xml` in diesem Beispiel wird verwendet, da das Stylesheet sich im Root-Ordner der Web-Server-Software befindet. Verwenden Sie für das Stylesheet einen URL mit einem absoluten Pfad, um die Position auf dem Web-Server anzugeben. Das Stylesheet kann sich auch auf einem anderen Web-Server befinden.

Auf der Grundlage dieser Abfrage nimmt die Web Publishing Engine die folgende Verarbeitungsanweisung in das XML-Dokument auf:

```
<?xml–stylesheet type="text/xsl" href="/mystylesheet.xml"?>
```

Kopieren oder platzieren Sie das Stylesheet für Client-seitige Verarbeitung auf den Web-Server an die Stelle, die der absolute Pfad im URL für das `HREF`-Attribut angibt.

Wichtig Platzieren Sie Stylesheets für Client-seitige Verarbeitung nicht in den Ordner `xslt-template-files`, der für Server-seitige XSLT-Stylesheet-Verarbeitung verwendet wird. Weitere Informationen finden Sie unter „Verwenden von FileMaker XSLT-Stylesheets in einer Website oder einem Programm“ auf Seite 32.

Hinweis Einige Webbrowser unterstützen keine Client-seitige Verarbeitung. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Webbrowser.

Fehlerbehebung beim XML-Dokumentzugriff

Wenn beim Zugriff auf XML-Dokumente mit der Web Publishing Engine Probleme auftreten, prüfen Sie Folgendes:

- Die erweiterten Zugriffsrechte in der Datenbank müssen auf „XML Custom Web Publishing“ eingestellt und einem Benutzerkonto zugewiesen sein. Weitere Informationen finden Sie unter „Aktivieren von Instant Web Publishing in einer Datenbank“ auf Seite 21.
- Die Datenbank wird auf der Datenbank-Server-Komponente des FileMaker Server-Einsatzes bereitgestellt und von FileMaker Server geöffnet. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
- Prüfen Sie, ob der verwendete Datenbankkontoname und ggf. Ihr Passwort korrekt sind.
- Die Web-Server-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- Die Web Publishing Engine-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- XML Publishing ist in der Web Publishing Engine-Komponente aktiviert. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Kapitel 6

Entwickeln von FileMaker XSLT-Stylesheets

Dieses Kapitel enthält Informationen über den Aufbau von FileMaker XSLT-Stylesheets und die Verwendung der FileMaker XSLT-Erweiterungsfunktionen.

Verwenden von XSLT-Stylesheets mit der Web Publishing Engine

Beachten Sie beim Entwickeln und Verwenden von XSLT-Stylesheets für die Abfrage von FileMaker XML-Daten über die Web Publishing Engine die folgenden Punkte:

- Um ein XSLT-Stylesheet mit der Web Publishing Engine zu verwenden, müssen Sie den Namen eines XSLT-Stylesheets in einem URL angeben. Wenn Sie kein Stylesheet angeben oder die Web Publishing Engine das Stylesheet nicht findet oder analysiert („parst“), zeigt die Web Publishing Engine eine Fehlerseite an. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 56.
- Der Stylesheet-Dateiname und der Name des Ordners, in dem das Stylesheet gespeichert ist, müssen URL-kodiertes UTF-8 sein. Falls Ihr Stylesheet mit älteren Webbrowsern kompatibel sein muss, beschränken Sie die Namen auf ASCII-Zeichen.
- Sie müssen die zu verwendende FileMaker XML-Grammatik angeben, entweder als Query-Parameter im URL oder als statisch definierten Query-Parameter in der Verarbeitungsanweisung `<?xslt-cwp-query?>`. Wenn Sie keine XML-Grammatik angeben, zeigt die Web Publishing Engine einen Fehler an. Weitere Informationen finden Sie unter „Angabe einer XML-Grammatik für ein FileMaker XSLT-Stylesheet“ auf Seite 58.
- Sie können die Query-Parameter zur Identifikation der angeforderten FileMaker XML-Daten im URL oder als statisch definierten Query-Parameter in der Verarbeitungsanweisung `<?xslt-cwp-query?>` angeben. Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 56 und „Verwenden von statisch definierten Query-Befehlen und Query-Parametern“ auf Seite 59.
- Optional können Sie die Textkodierung einer XSLT-Abfrage mithilfe des Query-Parameters `-encoding` angeben. Wenn Sie keine Kodierung angeben, verwendet die Web Publishing Engine ihre Standardkodierungseinstellung für Abfragen. Weitere Informationen finden Sie unter „Festlegen von Textkodierung für Abfragen“ auf Seite 61.
- Sie können optional eine Ausgabemethode über das `method`-Attribut des Elements `<xsl:output>` angeben. Wenn Sie keine Ausgabemethode angeben, verwendet die Web Publishing Engine HTML als Ausgabe. Sie können optional die Kodierung der Ausgabeseite über das `encoding`-Attribut des Elements `<xsl:output>` angeben. Wenn Sie keine Kodierung angeben, verwendet die Web Publishing Engine die Standardkodierungseinstellung für Ausgabeseiten. Weitere Informationen finden Sie unter „Angabe einer Ausgabemethode und -kodierung“ auf Seite 62.
- Sie können optional für E-Mail-Nachrichten, die von der Web Publishing Engine gesendet werden, die Textkodierung über einen Funktionsparameter für die Erweiterungsfunktion `fmxslt:send_email()` angeben. Weitere Informationen finden Sie unter „Senden von E-Mail-Nachrichten von der Web Publishing Engine“ auf Seite 71.

Den Aufbau einer Abfrage beginnt die Web Publishing Engine mit einem beliebigen Query-Befehl und Query-Parametern, die statisch in der optionalen Verarbeitungsanweisung `<?xslt-cwp-query?>` gespeichert sind. Der statisch definierte Query-Befehl und seine Parameter werden zur Basisabfrage. Die Verarbeitungsanweisung `<?xslt-cwp-query?>` ist in einem Stylesheet nicht erforderlich, aber ihre Basisabfrage hat Vorrang vor entsprechenden Query-Befehlen oder -Parametern, die im URL-Query-String angegeben sind. Die Web Publishing Engine fügt dann der Basisabfrage im URL-Query-String einen Query-Befehl oder zusätzliche Parameter hinzu, die *nicht* in der Verarbeitungsanweisung `<?xslt-cwp-query?>` definiert sind. Die Web Publishing Engine bezieht anhand dieser Abfrage die FileMaker XML-Daten und gibt sie in der angegebenen Ausgabemethode bzw. als HTML an Ihren Webbrowser oder Ihr Programm zurück.

Referenz der FileMaker XSLT-Erweiterungsfunktionen

Diese Version enthält eine FileMaker-Datenbank mit dem Namen „XSLT Reference.fp7“, die kurze Beschreibungen und Beispiele für jede FileMaker XSLT-Erweiterungsfunktion bietet. Die Funktionsreferenzdatenbank finden Sie im folgenden Verzeichnis auf jedem Rechner Ihres FileMaker Server-Einsatzes (Master- oder Arbeitscomputer).

Mac OS

/Library/FileMaker Server/Example/XSLT

Windows

<Laufwerk>:\Programme\FileMaker\FileMaker Server\Examples\XSLT

wobei <Laufwerk> das Hauptlaufwerk ist, von dem das System startet.

Über die FileMaker XSLT-Starter-Lösung

Diese Version umfasst eine FileMaker XSLT-Starter-Lösung, die ein Beispiel für die Möglichkeiten von XSLT-Lösungen aufzeigt. Die XSLT-Starter-Lösung finden Sie im folgenden Verzeichnis auf jedem Rechner Ihres FileMaker Server-Einsatzes (Master- oder Arbeitscomputer).

Mac OS

/Library/FileMaker Server/Example/XSLT/Starter Solution

Windows

<Laufwerk>:\Programme\FileMaker\FileMaker Server\Examples\XSLT\Starter Solution

wobei <Laufwerk> das Hauptlaufwerk ist, von dem das System startet.

Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets

Die URL-Syntax für die Verwendung von FileMaker XSLT-Stylesheets mit der Web Publishing Engine:

`<schema>://<host>[:<port>]/fmi/xsl/[<pfad>]/<stylesheet.xml>[?<query_string>]`

Dabei gilt:

- <scheme> kann das HTTP- oder HTTPS-Protokoll sein.
- <host> ist die IP-Adresse oder der Domänenname des Hosts, auf dem der Web-Server installiert ist.
- <port> ist optional und gibt den Port an, den der Web-Server überwacht. Ohne Portangabe wird der Standard-Port des Protokolls (Port 80 für HTTP oder Port 443 für HTTPS) verwendet.

- `<path>` ist optional und gibt den/die Ordner innerhalb des Ordners `xslt-template-files` an, in dem sich das XSLT-Stylesheet befindet.
- `<stylesheet.xml>` ist der Dateiname des XSLT-Stylesheets.
- `<query_string>` kann eine Kombination aus einem Query-Befehl und einem oder mehreren Query-Parametern für Custom Web Publishing mit XSLT sein. Weitere Informationen finden Sie unter „Verwenden von Query-Strings in FileMaker XSLT-Stylesheets“ auf Seite 58, und in Anhang A, „Gültige Namen in Query-Strings“. Wenn das angegebene Stylesheet eine `<?xslt-cwp-query?>`-Verarbeitungsanweisung enthält, haben der statisch definierte Query-Befehl und die Parameter Vorrang vor anderen Query-Befehlen oder Parametern im URL-Query-String. Weitere Informationen finden Sie unter „Verwenden von statisch definierten Query-Befehlen und Query-Parametern“ auf Seite 59.

Hinweis Die URL-Syntax einschließlich der Namen von Query-Befehl und -Parametern unterscheidet Groß- und Kleinschreibung, mit Ausnahme von Teilen des Query-Strings. Der URL ist vorwiegend in Kleinbuchstaben mit Ausnahme der Grammatiknamen in Großbuchstaben: `FMPXMLRESULT` und `FMPXMLLAYOUT`. Informationen über die Regeln für Groß-/Kleinschreibung im Query-String finden Sie unter „Richtlinien für die Verwendung von Query-Befehlen und -Parametern“ auf Seite 94.

Beispiel eines URL für die Verwendung eines FileMaker XSLT-Stylesheets mit der Web Publishing Engine:
http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xml?–grammar=fmresultset&–db=mydatabase&–lay=mylayout&–findall

Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen

In einem generierten XML-Dokument für eine XSLT-Lösung unterscheidet sich die Syntax für den Verweis auf ein Medienobjekt für Medienfelder, die das tatsächliche Objekt in der Datenbank speichern, und für Medienfelder, die einen Verweis auf das Objekt speichern.

- Wenn ein Medienfeld das tatsächliche Objekt in der Datenbank speichert, dann verwendet das `<data>`-Element des Medienfelds die folgende URL-Syntax für den Verweis auf das Objekt:

```
<data>/fmi/xsl/cnt/data.<extension>?<query string></data>
```

Dabei gilt: `<extension>` ist die Dateinamenerweiterung zur Identifizierung des Objekttyps wie beispielsweise `.jpg` oder `.mov`. Informationen über `<query string>` finden Sie im vorherigen Abschnitt, „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“.

Beispiel:

```
<data>/fmi/xsl/cnt/data.jpg?–db=products&–lay=sales&–field=product_image(1)&–recid=2</data>
```

Hinweis In der generierten XML für ein Medienfeld ist der Wert für den Query-Parameter `–field` ein vollständig qualifizierter Feldname. Die Zahl in Klammern gibt die Wiederholungsnummer für das Medienfeld an und wird für Wiederhol- und für Einzelfelder generiert. Weitere Informationen finden Sie unter „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 95.

Verwenden Sie zum Abrufen der Mediendaten aus der Datenbank die folgende Syntax:

```
<schema>://<host>[:<port>]/fmi/xsl/cnt/data.<erweiterung>?<query_string>
```

Informationen über `<schema>`, `<host>` oder `<port>` finden Sie im vorherigen Abschnitt, „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“.

Beispiel:

```
http://www.company.com/fmi/xsl/cnt/data.jpg?–db=products&–lay=sales&–field=product_image(1)&–recid=2
```

- Wenn ein Medienfeld einen Dateiverweis anstelle eines tatsächlichen Objekts speichert, enthält das <data>-Element des Medienfelds einen relativen Pfad, der auf das Objekt verweist. Wenn sich z. B. logo.jpg im Web-Ordner innerhalb des FileMaker Pro-Ordners befindet, lautet das <data>-Element des Medienfelds:

```
<data>/images/logo.jpg</data>
```

Hinweis Das Medienobjekt, auf das verwiesen wird, muss im Ordner FileMaker Pro Web gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird, und dann in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software kopiert oder verschoben werden. Weitere Informationen finden Sie unter „Veröffentlichen des Inhalts von Medienfeldern im Web“ auf Seite 23.

- Für ein leeres Medienfeld ist auch das <data>-Element leer.

Verwenden von Query-Strings in FileMaker XSLT-Stylesheets

Beim Verwenden eines Query-Strings in einem URL oder der Verarbeitungsanweisung <?xslt-cwp-query?> in einem FileMaker XSLT-Stylesheet können Sie beliebige Query-Befehle und -Parameter einschließen, die zur Abfrage von XML-Daten aus einer FileMaker-Datenbank definiert sind. Weitere Informationen finden Sie unter „Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings“ auf Seite 50.

Sie können auch den folgenden Query-Befehl und Parameter benutzen, die nur zur Verwendung mit FileMaker XSLT-Stylesheets definiert sind.

Verwenden Sie diesen XSLT Query-Befehl oder Parameternamen

Parametername	Ziel	Kommentar
-grammar3	Angeben der XML-Grammatik für XSLT-CWP-Abfragen oder für XSLT-Stylesheets. Weitere Informationen finden Sie im nächsten Abschnitt, „Angeben einer XML-Grammatik für ein FileMaker XSLT-Stylesheet“.	Dieser Query-Parameter ist in allen XSLT-Abfragen erforderlich.
-encoding	Angeben der Textkodierung für eine Abfrage. Weitere Informationen finden Sie unter „Festlegen von Textkodierung für Abfragen“ auf Seite 61.	Dieser Query-Parameter ist in allen XSLT-Abfragen optional.
-process	Verarbeitung eines Stylesheets ohne Datenabfrage. Weitere Informationen finden Sie unter „Verarbeitung von XSLT-Abfragen, die nicht FileMaker Server abfragen“ auf Seite 62.	Dieser Query-Befehl erfordert den Query-Parameter -grammar.
-token	Weitergeben von Werten zwischen Seiten, ohne Sitzungen oder Cookies zu verwenden. Weitere Informationen finden Sie unter „Verwendung von Tokens zur Übergabe von Informationen zwischen Stylesheets“ auf Seite 63.	Dieser Query-Parameter ist in allen XSLT-Abfragen optional.

Angeben einer XML-Grammatik für ein FileMaker XSLT-Stylesheet

Die empfohlene XML-Grammatik für Custom Web Publishing mit XSLT ist die Grammatik fmresultset, die zur bequemen Benutzung mit XSLT entwickelt wurde. Weitere Informationen finden Sie unter „Verwenden der fmresultset-Grammatik“ auf Seite 41. Sie können auch die älteren Grammatiken FMPXMLRESULT oder FMPXMLLAYOUT verwenden. Für den Zugriff auf Wertelisten und Feldanzeigedaten in Layouts müssen Sie die FMPXMLLAYOUT-Grammatik verwenden. Weitere Informationen finden Sie unter „Verwenden anderer FileMaker XML-Grammatiken“ auf Seite 45. Die Grammatik FMPDSORESULT kann nicht mit Custom Web Publishing mit XSLT verwendet werden.

Um die Grammatik für ein FileMaker XSLT-Stylesheet anzugeben, verwenden Sie den Query-Parameter -grammar in einem URL oder den statisch definierten Query-Parameter in der Verarbeitungsanweisung <?xslt-cwp-query?>.

In einem URL:

```
http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xml?-grammar=fmresultset&-db=mydatabase
&-lay=mylayout&-findall
```

In einer Verarbeitungsanweisung:

```
<?xslt-cwp-query params="-grammar=fmresultset&-db=mydatabase&-lay=mylayout&-findall"?>
```

Wichtig Wenn Sie keine XML-Grammatik für ein FileMaker XSLT-Stylesheet angeben, wird der Fehler QUERY -ER0001 angezeigt. Weitere Informationen finden Sie unter Anhang B, „Fehlercodes für Custom Web Publishing“.

Erläuterung von Namespaces und Präfixen für FileMaker XSLT-Stylesheets

Mithilfe eindeutiger XSLT-Namespaces können Sie XSLT-Tags der richtigen Anwendung zuordnen. Deklarieren Sie im Element `<xsl:stylesheet>` am Beginn aller FileMaker XSLT-Stylesheets die Namespaces für die FileMaker XSLT-Erweiterungsfunktionen und die jeweiligen Grammatiken, die Sie in dem Stylesheet verwenden.

Verwendung	Namespace deklarieren	Präfix verwenden
fmresultset-XML-Grammatik	<code>xmlns:fmrs="http://www.filemaker.com/xml/fmresultset"</code>	fmrs
FMPXMLRESULT-Grammatik	<code>xmlns:fmp="http://www.filemaker.com/fmpxmlresult"</code>	fmp
FMPXMLLAYOUT-Grammatik	<code>xmlns:fml="http://www.filemaker.com/fmpxmllayout"</code>	fml
Für die Query-XML-Grammatik	<code>xmlns:fmq="http://www.filemaker.com/xml/query"</code>	fmq
Für die FileMaker XSLT-Erweiterungsfunktionen	<code>xmlns:fmxslt="xalan://com.fmi.xslt.ExtensionFunctions"</code>	fmxslt

Sie müssen auch den folgenden erforderlichen Namespace in jedem FileMaker XSLT-Stylesheet deklarieren:

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
```

Beispiel für Deklarationen von Namespaces:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset"
  xmlns:fml="http://www.filemaker.com/fmpxmllayout"
  xmlns:fmq="http://www.filemaker.com/xml/query"
  xmlns:fmxslt="xalan://com.fmi.xslt.ExtensionFunctions"
  exclude-result-prefixes="xsl fmrs fmq fml fmxslt">
```

Verwenden von statisch definierten Query-Befehlen und Query-Parametern

Sie können mit Ihrem FileMaker XSLT-Stylesheet die unbefugte Verwendung von Query-Befehlen und Query-Parametern verhindern, indem Sie die zu verwendenden Query-Befehle und -Parameter beim Abfragen von XML-Daten statisch definieren. Wenn Query-Befehle und -Parameter statisch in einem Stylesheet definiert sind (was nicht erforderlich ist), haben sie Vorrang vor etwaigen übereinstimmenden Query-Befehlen oder -Parametern, die ein URL-Query-String angibt.

Die Stylesheets, die von dem XSLT-Site-Assistenten generiert werden, verwenden statisch definierte Query-Befehle und -Parameter. FileMaker empfiehlt die Verwendung statisch definierter Query-Befehle und -Parameter als beste Technik für eine größere Sicherheit Ihrer Lösung.

Um Query-Befehle und -Parameter statisch zu definieren, verwenden Sie die folgende Verarbeitungsanweisung am Beginn Ihres FileMaker XSLT-Stylesheets:

```
<?xslt-cwp-query params="query string-fragment"?>
```

Dabei gilt:

query string-fragment ist ein String mit den Name-Wert-Paaren in folgendem Format:

```
name=wert&name2=wert2....
```

Dabei gilt:

name ist ein String, der den Namen eines Query-Befehls, Query-Parameters oder Datenbankfelds angibt.

wert ist ein Stringwert beliebiger Länge. Verwenden Sie für Query-Parameter und Feldnamen den jeweiligen Wert, den Sie definieren wollen, z. B. `-db=Produkte`. Geben Sie für Query-Befehle kein `=`-Zeichen oder einen Wert nach dem Befehlsnamen wie `-findall` an. Weitere Informationen finden Sie unter Anhang A, „Gültige Namen in Query-Strings“.

Die im Fragment verwendeten Strings müssen URL-kodiert sein. Weitere Informationen finden Sie unter „Erläuterung von URL-Textkodierung“ auf Seite 39. Sie müssen dieselbe Zeichenkodierung verwenden, die das `encoding`-Attribut im Tag `<xsl:output>` angibt. Wenn keine Kodierung angegeben wird, verwendet die Web Publishing Engine ihre konfigurierte Standardkodierung.

Das Trennzeichen zwischen den beiden Name-Wert-Paaren muss ein kaufmännisches Und-Zeichen (`&`) sein. Nehmen Sie z. B. an, Sie haben die folgende Verarbeitungsanweisung in einem Stylesheet mit dem Namen `my_stylesheet.xml` verwendet:

```
<?xslt-cwp-query params="-db=products&-lay=sales&-grammar=fmresultset&productname=the%20item&-find"?>
```

Diese Beispiel-Verarbeitungsanweisung würde alle Abfragen an `my_stylesheet.xml` zwingen, die `fmresultset`-Grammatik mit der Datenbank `products` und dem Layout `sales` und eine `-find`-Abfrage mit dem Wert `the%20item` im Feld „`productname`“ zu verwenden.

Nun führt ein Client die folgende Abfrage mit `my_stylesheet.xml` durch:

```
http://server.company.com/fmi/xsl/my_stylesheet.xml?-lay=revenue&city=London&-edit
```

Die Web Publishing Engine verarbeitet in diesem Fall die folgende XML-Abfrage:

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=products&-lay=sales&productname=the%20item&city=London&-find
```

Der statisch definierte Query-Befehl und seine Parameter setzen den Query-Parameter `-lay=revenue` und den Query-Befehl `-edit` außer Kraft, die der Client angegeben hat. Da das Feld „`city`“ in der Verarbeitungsanweisung nicht statisch definiert war, übernimmt die Web Publishing Engine dafür in der XML-Abfrage den Wert `London`, den der Client eingegeben hat.

Festlegen von Textkodierung für Abfragen

Die Web Publishing Engine führt die folgenden Schritte in der gezeigten Reihenfolge aus, bis sie die Kodierung einer XSLT-Abfrage ermittelt:

1. Prüfen, ob das charset-Attribut im Abfrage-Header „Content-Type“ gesetzt ist.
2. Prüfen, ob Sie eine Kodierung mit dem Query-Parameter `-encoding` angegeben haben. Sie können diesen Parameter in einem URL oder als statisch definierten Query-Parameter in der Verarbeitungsanweisung `<?xslt-cwp-query?>` angeben. Der Wert des `-encoding`-Parameters gibt die Kodierung an, die für die übrigen Parameter in der Abfrage verwendet wird. Die gültigen Werte für diesen Parameter werden in der folgenden Tabelle aufgelistet. Beispiel:
`http://192.168.123.101/fmi/xsl/template/my_stylesheet.xml?-db=products-lay=sales&-grammar=fmresultset
&-encoding=Shift_JIS&-findall`
3. Verwenden der aktuellen Einstellung für die Standard-Textkodierungsoption der Abfrage- und Ausgabeseiten für die Web Publishing Engine. Direkt nach der Installation der Web Publishing Engine ist die Textkodierung für Abfragen standardmäßig auf UTF-8 eingestellt. Sie können die Textkodierungseinstellungen der Web Publishing Engine über Admin Console ändern. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Nachdem die Web Publishing Engine die Kodierung ermittelt, wird diese Kodierung verwendet und es werden keine weiteren Schritte zur Ermittlung der Kodierung unternommen. Wenn z. B. das charset-Attribut im Abfrage-Header „Content-Type“ gesetzt ist, verwendet die Web Publishing Engine nicht den Wert des Query-Parameters `-encoding`.

Die Textkodierung, die über eine der obigen Methoden angegeben wird, muss eine der folgenden Kodierungen verwenden:

Kodierung	Beschreibung
US-ASCII	Der ASCII-Basiszeichensatz wird typischerweise für englische E-Mails mit Standardtext verwendet.
ISO-8859-1	Der Latin 1-Zeichensatz wird typischerweise für Web-Seiten und E-Mail-Nachrichten mit lateinischen Buchstaben aus dem oberen ASCII-Zeichenbereich verwendet.
ISO-8859-15	Der Latin 9-Zeichensatz, der beinahe mit dem Latin 1-Zeichensatz übereinstimmt, umfasst zusätzlich das Euro-Symbol €.
ISO-2022-JP	Die ISO Japanese-Kodierung wird typischerweise für japanische E-Mail-Nachrichten verwendet.
Shift_JIS	Diese Japanisch-Kodierung wird typischerweise für japanische Web-Seiten verwendet.
UTF-8	Die Acht-Bit-Kodierung von Unicode. Die Verwendung von UTF-8 für E-Mail-Nachrichten und Web-Seiten gewinnt an Beliebtheit, da gängige Browser und E-Mail-Clients diese Kodierung jetzt unterstützen. Da UTF-8 den kompletten Bereich an Unicode-Zeichen unterstützt, sind damit Seiten in jeder Sprache möglich.

Hinweise

- Direkt nach der Installation der Web Publishing Engine ist die Textkodierung für Ausgabeseiten standardmäßig auf UTF-8 eingestellt. Weitere Informationen finden Sie im nächsten Abschnitt, „Angabe einer Ausgabemethode und -kodierung“. Für E-Mail-Nachrichten verwendet die Web Publishing Engine anfangs für die Textkodierung ISO-8859-1 als Standardeinstellung. Sie können diese Einstellung über Admin Console ändern.
- Sie können auch die E-Mail-Nachrichtenkodierung mithilfe der Erweiterungsfunktion `fmxslt:send_email(String smtpFields, String body, String encoding)` festlegen. Weitere Informationen finden Sie unter „Senden von E-Mail-Nachrichten von der Web Publishing Engine“ auf Seite 71.

Angeben einer Ausgabemethode und -kodierung

Sie können eine Ausgabemethode und -kodierung der Ausgabeseiten mithilfe des `method`- und `encoding`-Attributs des Elements `<xsl:output>` angeben. Beide Attribute sind optional.

Das `method`-Attribut legt den Typ der Ausgabe fest, d. h. „html“, „text“ oder „xml“. Andere Methodentypen werden nicht unterstützt. Wenn Sie keine Methode angeben, verwendet die Web Publishing Engine „html“.

Das `encoding`-Attribut gibt die Kodierung von Ausgabeseiten an. Sie können eine beliebige Kodierung angeben, die in der Tabelle im vorherigen Abschnitt aufgeführt ist. Wenn Sie keine Kodierung angeben, verwendet die Web Publishing Engine ihre Standardkodierungseinstellung für Ausgabeseiten.

Beispiel:

```
<xsl:output method="html" encoding="ISO-8859-1"/>
```

Wenn Sie das Element `<xsl:output>` nicht in einem Stylesheet verwenden, gibt die Web Publishing Engine HTML-Seiten in der aktuellen Standardeinstellung für Textkodierung von Ausgabeseiten aus.

Erläuterung der Kodierung von XSLT-Stylesheets

Neben der Kodierung für Abfragen und Ausgabeseiten muss die Kodierung Ihrer XSLT-Stylesheets auch im `encoding`-Attribut der XML-Deklaration am Beginn des Stylesheets angegeben werden. Sie können eine beliebige Kodierung angeben, aufgeführt in der Tabelle auf Seite 61.

Beispielsweise gibt die folgende Deklaration UTF-8 als Kodierung des Stylesheets an:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Wenn Sie keine Stylesheet-Kodierung angeben, nimmt die Web Publishing Engine UTF-8 als Kodierung an.

Verarbeitung von XSLT-Abfragen, die nicht FileMaker Server abfragen

Sie können mithilfe des Query-Befehls `–process` XSLT-Abfragen verarbeiten, wenn diese keine Daten aus der Datenbank benötigen oder wenn Ihr Stylesheet keine datenbankspezifischen Informationen wie Datensätze, Feldnamen oder Layoutnamen erfordert. Indem Sie in diesen Fällen den Befehl `–process` verwenden, können Sie die Arbeitslast von FileMaker Server verringern.

Mithilfe des Befehls `–process` können Sie z. B.:

- ein Stylesheet laden, das eine statische Seite generiert, wenn keine Datenbankinformation benötigt wird.
- ein Stylesheet laden, das einen neuen Datensatz erstellt, wenn das Stylesheet keine Datenbank- oder Layoutinformation wie eine Werteliste benötigt.
- eine Erweiterungsfunktion wie `fmxml:send_email()` verwenden, die keine Daten aus der Datenbank benötigt.
- auf Informationen zugreifen, die in einer Sitzung gespeichert sind, wenn keine Datenbankinformation benötigt wird.

Der Befehl `–process` gibt ein XML-Dokument zurück, das Produktinformationen über die Web Publishing Engine enthält.

Der einzige erforderliche Parameter für den Befehl `–process` ist `–grammar` und Sie müssen die `fmresultset`- oder die `FMPXMLRESULT`-Grammatik verwenden.

Beispiel:

```
http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xml?–grammar=fmresultset&–process
```

Verwendung von Tokens zur Übergabe von Informationen zwischen Stylesheets

Sie können den Query-Parameter `-token` in einem URL oder als statisch definierten Query-Befehl verwenden, um beliebige benutzerdefinierte Informationen ohne Sitzungen oder Cookies zwischen Stylesheets zu übergeben. Der Query-Parameter `-token` ist in allen Query-Befehlen optional.

Der benutzerdefinierte Parameterwert kann ein beliebiger URL-kodierter String sein. Beispiel:

```
http://192.168.123.101/fmi/xsl/template/my_stylesheet.xml?-db=products&-lay=sales&-grammar=fmresultset
&-token.D100=Pending&-findall
```

Weitere Informationen finden Sie unter „Query-Parameter `-token`.[string] (Werte zwischen XSLT-Stylesheets übergeben)“ auf Seite 111.

Wichtig Verwenden Sie den Query-Parameter `-token` nicht, um private Daten zu übergeben.

Um den Wert des Query-Parameters `-token` abzurufen, verwenden Sie die Anweisung `<xsl:param name="request-query"/>`. Weitere Informationen finden Sie unter „Verwenden der Query-Information in einer Abfrage“ auf Seite 64.

Verwenden der FileMaker XSLT-Erweiterungsfunktionen und -parameter

Die FileMaker XSLT-Erweiterungsfunktionen befinden sich laut Definition im `fmxml:namespace`. Stellen Sie sicher, dass Sie eine Deklaration des `fmxml:namespace` im Element `<xsl:stylesheet>` am Beginn Ihres XSLT-Stylesheets angeben. Weitere Informationen finden Sie unter „Erläuterung von Namespaces und Präfixen für FileMaker XSLT-Stylesheets“ auf Seite 59.

Die FileMaker XSLT-Erweiterungsfunktionen wurden so entwickelt, dass Sie sie in einem XSLT-Stylesheet verwenden können, indem Sie sie als Funktionsaufruf in einer XPath-Anweisung angeben. XPath-Anweisungen werden als Werte des `select`-Attributs und des `test`-Attributs in zahlreichen XSLT-Elementen verwendet.

Nehmen Sie z. B. an, Sie wollen im User-Agent-Header feststellen, welcher Browser benutzt wird. Dafür sollten Sie eine Variable verwenden, die den Wert des User-Agent-Headers enthält:

```
<xsl:variable name="user-agent" select="fmxml:get_header('User-Agent')"/>
```

Für die Erweiterungsfunktionen, die einen Wert zurückgeben, weist der Wert den angegebenen XSLT-Typ auf. Viele Funktionen geben Strings zurück, aber einige Funktionen geben eine Knotengruppe (`node-set`) zurück, die durchlaufen werden kann.

Hinweis Dieser Abschnitt beschreibt die FileMaker XSLT-Erweiterungsfunktionen und -parameter und bietet einige Beispiele. Zusätzliche Beispiele für jede Funktion finden Sie in der Referenz der FileMaker XSLT-Erweiterungsfunktionen. Weitere Informationen finden Sie unter „Referenz der FileMaker XSLT-Erweiterungsfunktionen“ auf Seite 56.

Erläuterung der FileMaker-spezifischen XSLT-Parameter, die von der Web Publishing Engine gesetzt werden

Beim Verarbeiten einer Abfrage setzt die Web Publishing Engine dynamisch die Werte der folgenden FileMaker-spezifischen XSLT-Parameter. Sie können die Werte dieser Parameter in Ihrem Stylesheet mithilfe des Elements `<xsl:param>` verwenden.

FileMaker-spezifischer XSLT-Parameter	Weitere Informationen
<code><xsl:param name="request-query"/></code>	„Verwenden der Query-Information in einer Abfrage“ im nächsten Abschnitt.
<code><xsl:param name="client-ip"/></code> <code><xsl:param name="client-user-name"/></code> <code><xsl:param name="client-password"/></code>	„Abrufen von Clientinformationen“ auf Seite 65.
<code><xsl:param name="xml-base-uri"/></code>	„Verwenden des Basis-URI-Parameters der Web Publishing Engine“ auf Seite 65.
<code><xsl:param name="authenticated-xml-base-uri"></code>	„Verwenden des Parameters „authenticated base URI““ auf Seite 66.

Verwenden der Query-Information in einer Abfrage

Sie können mithilfe eines FileMaker XSLT-Parameters auf Informationen in einer Abfrage in einem URL oder in HTML-Formulardaten zugreifen. Beispielsweise können Sie auf die aktuelle Abfrage-Query-Information zugreifen, um die aktuelle Position in einer Datensatz-Ergebnismenge zu bestimmen und Verknüpfungen zum vorherigen und nächsten Datensatz zu erstellen.

Die folgenden FileMaker XSLT-Parameter bieten Zugriff auf alle Query-Befehle und Query-Parameter, die für Abfragen von FileMaker XML-Daten über die Web Publishing Engine verwendet werden:

```
<xsl:param name="request-query"/>
```

Mit Ausnahme von Feldnamen gibt die Web Publishing Engine alle Namen von Query-Befehlen und Query-Parametern in Kleinbuchstaben zurück. Die Groß-/Kleinschreibung von Feldnamen wird beibehalten.

Ein XML-Dokumentfragment wird in den Abfrage-Query-Parameter in der folgenden Grammatik geladen:

```
<!DOCTYPE query [
  <!ELEMENT query (parameter)*>
    <!ATTLIST query action CDATA #REQUIRED>
    <!ELEMENT parameter (#PCDATA)>
    <!ATTLIST parameter name CDATA #REQUIRED>
  ]
```

Hinweis Die Query-Information befindet sich laut Definition im Namespace `fmq="http://www.filemaker.com/xml/query"`. Stellen Sie sicher, dass Sie eine Deklaration des `fmq`-Namespace im Element `<xsl:stylesheet>` am Beginn Ihres XSLT-Stylesheets angeben. Weitere Informationen finden Sie unter „Erläuterung von Namespaces und Präfixen für FileMaker XSLT-Stylesheets“ auf Seite 59.

Nehmen Sie z. B. an, Sie wollen auf die Query-Befehle und Query-Parameter in dieser Abfrage zugreifen:

```
http://192.168.123.101/fmi/xsl/my_stylesheets.xml?-db=products&-lay=sales&-grammar=fmresultset&-token.1=abc123&-findall
```

Wenn Sie die Anweisung `<xsl:param name="request-query" />` vor dem Vorlagenabschnitt („template“) einfügen, speichert die Web Publishing Engine dieses XML-Dokumentfragment in diesem Parameter:

```
<query action="my_stylesheet.xml" xmlns="http://www.filemaker.com/xml/query">
  <parameter name="-db">products</parameter>
  <parameter name="-lay">sales</parameter>
  <parameter name="-grammar">fmresultset</parameter>
  <parameter name="-token.1">abc123</parameter>
  <parameter name="-findall"></parameter>
</query>
```

Sie können dann den Parameter `request-query` verwenden, um auf den Wert eines Tokens zuzugreifen, der über einen XPath-Ausdruck in einem URL übergeben wurde. Beispiel:

```
$request-query/fmq:query/fmq:parameter[@name = '-token.1']
```

Abrufen von Clientinformationen

Sie können mithilfe der folgenden FileMaker XSLT-Parameter Informationen über die IP-Adresse, den Benutzernamen und das Passwort eines Web-Clients von der Web Publishing Engine abrufen:

```
<xsl:param name="client-ip"/>
<xsl:param name="client-user-name"/>
<xsl:param name="client-password">
```

Nehmen Sie diese Parameteranweisungen vor dem ersten `<xsl:template>`-Element in Ihr XSLT-Stylesheet auf.

Diese Parameter identifizieren den Web-Benutzer, wenn ein Stylesheet programmtechnisch weitere passwortgeschützte XML-Dokumente lädt. Weitere Informationen finden Sie unter „Laden zusätzlicher Dokumente“ auf Seite 66. Der Web-Benutzer muss den Benutzernamen und das Passwort anfangs im Dialogfeld „HTTP-Standardauthentifizierung“ angeben. Weitere Informationen finden Sie unter „Zugriff auf eine geschützte Datenbank“ auf Seite 21.

Weitere Informationen und Beispiele für die Verwendung dieser drei FileMaker XSLT-Parameter finden Sie in der Referenz der FileMaker XSLT-Erweiterungsfunktionen.

Verwenden des Basis-URI-Parameters der Web Publishing Engine

Die Web Publishing Engine definiert den Basis-Uniform Resource Identifier (URI)-Parameter als Host und Port, an dem die Web Publishing Engine installiert ist. Dieser Basis-URI ermöglicht, dass Abfragen von XML-Daten aus FileMaker-Datenbanken in Relation zum Web Publishing Engine-Host aufgelöst werden.

Um auf den Basis-URI der Web Publishing Engine zuzugreifen, nehmen Sie diese Anweisungen vor dem ersten `<xsl:template>`-Element in Ihr XSLT-Stylesheet auf:

```
<xsl:param name="xml-base-uri"/>
```

Sie können den Basis-URI dann für das aktuelle Stylesheet über die Variable `$xml-base-uri` immer dann verwenden, wenn Sie eine zusätzliche Abfrage auf FileMaker XML-Daten ausführen müssen. Sie können z. B. den Basis-URI in der folgenden Abfrage von zusätzlichen XML-Daten verwenden:

```
<xsl:variable name="layout_information" select="document(concat($xml-base-uri,'/fmi/xml/FMPXMMLLAYOUT.xml?
-db=products&-lay=sales&-view'))" />
```

Verwenden des Parameters „authenticated base URI“

Der Parameter `authenticated-xml-base-uri` kombiniert die Funktionalität der Parameter `client-user-name` und `client-password` mit dem Parameter `xml-base-uri`:

```
<xsl:param name="authenticated-xml-base-uri"/>
```

Verwenden Sie diesen Parameter, um ein zusätzliches, passwortgeschütztes XML-Dokument zu laden, das den Benutzernamen und das Passwort verlangt, das anfangs in der aktuell verarbeiteten Abfrage eingegeben wurde. Ein Beispiel finden Sie im nächsten Abschnitt, „Laden zusätzlicher Dokumente“.

Nehmen Sie diese Parameteranweisung vor dem ersten `<xsl:template>`-Element in Ihr XSLT-Stylesheet auf.

Wenn die Werte der Parameter `client-user-name` und `client-password` nicht leer sind, lautet der Wert des Parameters `authenticated-xml-base-uri`:

```
http://username:password@hostname:port
```

Wenn die Werte der Parameter `client-user-name` und `client-password` leer sind, ist der Wert des Parameters `authenticated-xml-base-uri` identisch mit dem Wert des Parameters `xml-base-uri`.

Laden zusätzlicher Dokumente

Um bei der Verarbeitung eines XSLT-Stylesheets ein zusätzliches XML-Dokument zu laden, verwenden Sie die XSLT-Standardfunktion `document()` mit einem URI für das XML-Dokument. Die Funktion `document()` gibt die angeforderte XML als eine Knotengruppe (`node-set`) zurück, die in einem `<xsl:variable>`-Element gespeichert werden kann.

Um ein XML-Dokument zu laden, das Daten aus einer FileMaker-Datenbank enthält, verwenden Sie die Funktion `document()` mit FileMaker-Query-Befehl und -Parametern. Beispiel:

```
<xsl:variable name="other-data" select="document(concat($xml-base-uri,'/fmi/xml/FMPXMMLLAYOUT.xml?
-db=products&-lay=sales&-view'))" />
```

Um ein zusätzliches, passwortgeschütztes XML-Dokument zu laden, das den Benutzernamen und das Passwort verlangt, das ursprünglich in der aktuell verarbeiteten Abfrage eingegeben wurde, verwenden Sie den Parameter `authenticated-xml-base-uri`. Dieser Parameter gibt denselben Benutzernamen und dasselbe Passwort als Teil des URI an, der an die Funktion `document()` übergeben wird.

Beispiel:

```
<xsl:variable name="other-data" select="document(concat($authenticated-xml-base-uri,
'/fmi/xml/FMPXMMLLAYOUT.xml?-db=products&-lay=sales&-view'))"/>
```

Um ein passwortgeschütztes XML-Dokument zu öffnen, das einen anderen Benutzernamen und ein anderes Passwort verlangt, als in der übergeordneten Abfrage angegeben wurde, geben Sie anhand der folgenden Syntax den Benutzernamen und das Passwort als Teil des URI an, der an die Funktion `document()` übergeben wurde:

```
http://username:password@hostname/path?querystring
```

Um ein XML-Dokument zu laden, das nicht auf einer FileMaker-Datenbank basiert, verwenden Sie die Funktion `document()` ohne FileMaker-Query-Befehle oder Query-Parameter. Beispiel:

```
<xsl:variable name="other-data" select="document('http://server.company.com/data.xml')" />
```

Wenn Sie die Funktion `document()` mit einem relativen URL verwenden, versucht die Web Publishing Engine, das XML-Dokument von der Position im lokalen Dateisystem zu laden, die dem relativen Speicherort des Stylesheets entspricht. Nehmen Sie z. B. an, dass ein Stylesheet, das sich im Ordner `mystylesheets` innerhalb des Ordners `xslt-template-files` befindet, die folgende `document()`-Funktion mit einem relativen URL enthält:

```
<xsl:variable name="mydoc" select="document('mystylesheets/mydoc.xml')"/>
```

Die Web Publishing Engine versucht, `mydoc.xml` aus dem Ordner `mystylesheets` im Ordner `xslt-template-files` im lokalen Dateisystem zu laden.

Hinweis Wenn Sie den Basis-URI der Web Publishing Engine verwenden, um ein Dokument zu laden, unterstützt die Web Publishing Engine nur HTTP. Wenn Sie ein Dokument von einem externen Server laden, unterstützt die Web Publishing Engine sowohl HTTP als auch HTTPS.

Verwenden der Layoutinformationen für eine Datenbank in einem Stylesheet

Sie können die Layoutinformation einer FileMaker-Datenbank in ein Stylesheet übernehmen, indem Sie die Information mit der `FMPXMLLAYOUT`-Grammatik abfragen und dann über die XSLT-Funktion `document()` in eine Variable laden:

```
<xsl:variable name="layout" select="document(concat($xml-base-uri,'/fmi/xml/FMPXMLLAYOUT.xml?–view'))"/>
```

Nehmen Sie z. B. an, Sie wollen ein Menü für ein Feld mit dem Namen `Color` erstellen, das mit den Werten aus einer `shirts` genannten Werteliste mit zwei Feldern gefüllt ist, die wiederum in einem Layout in einer FileMaker-Datenbank definiert ist. Nehmen wir an, das erste Feld in der Werteliste mit zwei Feldern speichert die ID-Nummer der Farbe (wie z. B. „100“) und das zweite Feld speichert den zugehörigen Namen der Farbe (wie z. B. „Hellgrün“). So können Sie die Funktion `document()` verwenden, um die Layoutinformationen in eine XSLT-Variable zusammen mit dem Attribut `DISPLAY` zu laden, um den Wert für das zweite Feld in einer Werteliste mit zwei Feldern anzuzeigen:

```
<xsl:variable name="layout" select="document(concat($xml-base-uri,'/fmi/xml/FMPXMLLAYOUT.xml?–db=products
&–lay=sales&–view'))"/>
```

```
<select size="1">
```

```
  <xsl:attribute name="name">Color</xsl:attribute>
```

```
  <option value="">Select One...</option>
```

```
  <xsl:for-each select="$layout/fmi:FMPXMLLAYOUT/fmi:VALUELISTS/fmi:VALUELIST[@NAME = 'shirts']/
fmi:VALUE">
```

```
    <option:>
```

```
      <xsl:attribute name="value"><xsl:value-of select="."/></xsl:attribute>
```

```
      <xsl:value-of select="@DISPLAY"/>
```

```
    </option>
```

```
  </xsl:for-each>
```

```
</select>
```

Verwenden von Inhaltspufferung

Wenn Inhaltspufferung deaktiviert ist, streamt die Web Publishing Engine das Ergebnis einer XSLT-Transformation direkt zurück an den Client. Inhaltspufferung ist immer deaktiviert, sofern Sie sie nicht explizit aktivieren. Wenn Sie Inhaltspufferung aktivieren, speichert die Web Publishing Engine den umgewandelten Inhalt, bis die vollständige Transformation beendet ist.

Inhaltspufferung wird für XSLT-Stylesheets benötigt, die Header manipulieren. Da Header vor dem Antworttext geschrieben werden, muss der Text gepuffert werden, damit die hinzugefügte Header-Information eingeschlossen werden kann.

Es gibt vier FileMaker-Erweiterungsfunktionen, für die das XSLT-Umwandlungsergebnis gepuffert werden muss:

- `fmxml:create_session()`: Weitere Informationen finden Sie unter „Verwenden der Sitzungserweiterungsfunktionen“ auf Seite 69.
- `fmxml:set_header()`: Weitere Informationen finden Sie unter „Verwenden von Header-Funktionen“ auf Seite 73.
- `fmxml:set_status_code ()`: Weitere Informationen finden Sie unter „Verwenden von Header-Funktionen“ auf Seite 73.
- `fmxml:set_cookie()`: Weitere Informationen finden Sie unter „Verwenden der Cookie-Erweiterungsfunktionen“ auf Seite 74.

Damit diese FileMaker-Erweiterungsfunktionen korrekt arbeiten, müssen Sie die folgende XSLT-Verarbeitungsanweisung für die Abfrage in das Dokument der obersten Ebene aufnehmen:

```
<?xslt-cwp-buffer buffer-content="true"?>
```

Wichtig Wenn ein Basis-Stylesheet ein weiteres Stylesheet enthält, muss das Basis-Stylesheet die Verarbeitungsanweisung `<?xslt-cwp-buffer?>` enthalten. Diese Anweisung wird ignoriert, wenn sie in einem eingeschlossenen Stylesheet verwendet wird.

Ein Vorteil der Verwendung der Verarbeitungsanweisung zur Pufferung der Antwort ist, dass die Web Publishing Engine die Länge der Antwort ermitteln und den Content-Length-Header in der Antwort setzen kann. Durch Puffern der Antwort kann die Leistung der Web Publishing Engine beeinträchtigt werden.

Verwenden von Web Publishing Engine-Sitzungen zum Speichern von Informationen zwischen Abfragen

Mithilfe der Server-seitigen Sitzungen der Web Publishing Engine können Sie jeden Informationstyp zwischen Abfragen verfolgen und speichern. Sitzungen erlauben Ihnen, eine Web-Anwendung zu erstellen, die mithilfe persistenter beliebiger Informationen den Status zwischen Abfragen beibehalten kann. So könnten beispielsweise Benutzer-Client-Informationen, die auf einer ersten Formularseite eingegeben wurden, in einer Sitzung gespeichert und dann zum Ausfüllen der Werte auf einer späteren Seite verwendet werden.

Standardmäßig speichert die Web Publishing Engine die Sitzungs-ID in einem Cookie. Um Clients zu berücksichtigen, die keine Cookies erlauben, können Sie dem URL die Sitzungs-ID mithilfe der Funktion `fmxml:session_encode_url()` hinzufügen. Damit in allen Situationen Kompatibilität gewährleistet ist, sollten Sie alle URLs auf der Seite mit der Funktion `fmxml:session_encode_url()` kodieren. Diese Funktion fügt Ihrem URL den durch Semikolons getrennten Parameter `jsessionId` hinzu, der die Kennung für die übergeordnete Sitzung des jeweiligen Clients darstellt.

Anstatt den folgenden Link auf eine Seite zu setzen:

```
<a href="my_stylesheets.xml?-db=products&-lay=sales&-grammar=fmresultset&-findall">hyperlinked text</a>
```

Sollten Sie alle Links wie folgt auf eine Seite platzieren:

```
<a href="{fmxslt:session_encode_url('my_stylesheet.xml?-db=products&-lay=sales&-grammar=fmresultset
&-findall')}">hyperlinked text</a>
```

Wenn der Client keine Cookies zulässt, beinhaltet die Seite:

```
<a href="my_stylesheet.xml;jsessionid=<session id>?-db=products&-lay=sales&-grammar=fmresultset&-findall">
hyperlinked text</a>
```

Wenn die Web Publishing Engine erkennt, dass der Client Cookies erlaubt, speichert die Funktion `fmxslt:session_encode_url()` die Sitzungs-ID in einem Cookie anstelle eines URLs.

Hinweis Sitzungsinformationen bleiben nach einem Neustart der Web Publishing Engine nicht erhalten.

Verwenden der Sitzungserweiterungsfunktionen

Verwenden Sie die folgenden Erweiterungsfunktionen für Sitzungen, um Sitzungsvariablen zu manipulieren. Sie können einen String, eine Zahl, einen booleschen Wert oder eine Knotengruppe (`node-set`) in einem Sitzungsobjekt speichern. Mithilfe einer Knotengruppe (`node-set`) können Sie eine Datenstruktur in XML anlegen und sie dann zwischen Abfragen im Sitzungsobjekt speichern.

Sitzungserweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
<code>fmxslt:session_exists(String Sitzungsname)</code>	Boolescher Wert	Prüft, ob eine Sitzung mit dem angegebenen Namen existiert.
<code>fmxslt:create_session(String Sitzungsname)</code>	Boolescher Wert	Erstellt eine Sitzung mit dem angegebenen Sitzungsnamen und der Standard-Zeitüberschreitung, die über Admin Console gesetzt wird. Weitere Informationen finden Sie in der FileMaker Server Hilfe. Hinweis Diese Funktion verlangt die Verarbeitungsanweisung <code><?xslt-cwp-buffer?></code> . Weitere Informationen finden Sie unter „Verwenden von Inhaltspufferung“ auf Seite 68.
<code>fmxslt:invalidate_session(String Sitzungsname)</code>	Boolescher Wert	Erzwingt eine sofortige Zeitüberschreitung der Sitzung.
<code>fmxslt:set_session_timeout(String Sitzungsname, Number Zeitüberschreitung)</code>	Boolescher Wert	Setzt die Sitzungszeitüberschreitung in Sekunden. Die Standardzeitüberschreitung für Sitzungen wird über Admin Console eingestellt.
<code>fmxslt:session_encode_url(String URL)</code>	String	Kodiert einen URL mit der Sitzungs-ID, wenn der Client keine Cookies unterstützt; gibt anderenfalls den eingegebenen URL zurück.

Sitzungserweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
fmxml:set_session_object(String Sitzungsname, String Name, Object Wert)	XSLT-Objekt (Zahl, String, Boolescher Wert oder Knotengruppe)	Speichert ein XSLT-Objekt (Zahl, String, Boolescher Wert oder Knotengruppe (node-set)) unter einer Sitzung, die später über die Funktion fmxml:get_session_object () abgerufen werden kann. Diese Funktion gibt auch das zuvor unter dem angegebenen Sitzungsobjektnamen gespeicherte Objekt zurück. Wenn kein Objekt unter dem Namen gespeichert wurde, wird ein Null-Objekt zurückgegeben. Hinweis Die Erweiterungsfunktion set_session_object() kann nur Zeichenfolgenwerte speichern und interpretiert an sie übergebene Objekte als Zeichenfolge. Wenn das Objekt nicht in eine Zeichenfolge konvertiert werden kann, ist kein Wert in der Sitzung gespeichert und der Fehlercode der Erweiterungsfunktion ist auf 10100 (Unbekannter Sitzungsfehler) gesetzt. Jeder Versuch, ein Sitzungsobjekt mit null oder einer leeren Zeichenfolge zu setzen, resultiert ebenfalls im Fehlercode 10100 (Unbekannter Sitzungsfehler). Um eine Sitzungsvariable zu löschen, entfernen Sie sie aus der Sitzung mithilfe der Funktion remove_session_object().
fmxml:get_session_object(String Sitzungsname, String Name)	XSLT-Objekt	Bezieht ein XSLT-Objekt aus der Sitzung.
fmxml:remove_session_object(String Sitzungsname, String Name)	XSLT-Objekt	Gibt ein XSLT-Objekt zurück und entfernt es dann aus der Sitzung.

Beispiel für das Erstellen einer Sitzung und anschließendes Speichern einer bevorzugten Farbe in der Sitzung:

```
<xsl:variable name="session">
  <xsl:choose>
    <xsl:when test="not (fmxml:session_exists(string($session-name)))">
      <xsl:value-of select="fmxml:create_session(string($session-name))"/>
    </xsl:when>
    <xsl:otherwise>true</xsl:otherwise>
  </xsl:choose>
</xsl:variable>
<xsl:variable name="favorite-color" select="fmxml:set_session_object(string($session-name), 'favorite-color', string($color))"/>
```

Wichtig

- Um sicherzustellen, dass Benutzer nach Abschluss einer Sitzung von einer Datenbank abgemeldet werden, erzwingen Sie anhand der Funktion fmxml:invalidate_session () eine sofortige Zeitüberschreitung der Sitzung.

- Wenn Sie Variablenfelder oder ein Script benutzen, das einen Status festlegt oder ändert, müssen Sie mithilfe von Admin Console die Option XSLT-Datenbanksitzungen für die Web Publishing Engine aktivieren. Anderenfalls werden Variablenfeldwerte und Status zwischen Abfragen nicht beibehalten. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
- Wenn Sie bei der Verwendung von Web Publishing Engine-Sitzungen in eine andere Datei wechseln, werden globale Feldwerte nicht beibehalten. Die Web Publishing Engine schließt die erste Datei, bevor sie die zweite Datei öffnet. Alternativ können Sie auf Daten aus der zweiten Datenbank zugreifen, indem Sie ein Layout in der ersten Datenbankdatei verwenden.

Senden von E-Mail-Nachrichten von der Web Publishing Engine

Mithilfe der Web Publishing Engine können Sie E-Mail-Nachrichten generieren, was für eigene Web-Lösungen nützlich ist. Damit die Web Publishing Engine eine E-Mail-Nachricht sendet, verwenden Sie eine der folgenden drei `fmxml:send_email()`-Erweiterungsfunktionen in einem XSLT-Stylesheet. Sie können diese Funktionen benutzen, um eine oder mehrere separate Nachrichten zu senden. Da die `fmxml:send_email()`-Funktionen im Server-seitigen XSLT-Stylesheet der Web Publishing Engine enthalten sind, kann ein Client keine unautorisierten E-Mail-Nachrichten über die Web Publishing Engine senden.

E-Mail-Erweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
<code>fmxml:send_email(String smtpFelder, String Nachrichtentext)</code>	Boolescher Wert	Sendet eine E-Mail-Nachricht beliebiger Länge in Standardtext aus der Web Publishing Engine und verwendet dafür die Standard-Textkodierung der Web Publishing Engine.
<code>fmxml:send_email(String smtpFelder, String Nachrichtentext, String Kodierung)</code>	Boolescher Wert	Sendet eine E-Mail-Nachricht beliebiger Länge in Standardtext und verwendet dafür eine der folgenden Textkodierungen: US-ASCII, ISO-8859-1, ISO-8859-15, ISO-2022-JP, Shift_JIS, UTF-8. Informationen u diesen Kodierungen finden Sie unter „Festlegen von Textkodierung für Abfragen“ auf Seite 61.
<code>fmxml:send_email(String smtpFelder, String xsltDatei, Node xml, boolean inklBilder)</code>	Boolescher Wert	Sendet eine HTML-basierte E-Mail-Nachricht mit der Zeichenkodierung, die über das <code>encoding</code> -Attribut des <code><xsl:output></code> -Elements im Stylesheet angegeben ist. Wenn das <code>encoding</code> -Attribut nicht im <code><xsl:output></code> -Element angegeben wurde, wird für E-Mail-Nachrichten die Standard-Textkodierung der Web Publishing Engine verwendet.

Hinweise

- In jeder der drei Formen der Funktion `fmxml:send_email()` ist der Parameter `smtpFields` ein URL-kodierter String beliebiger Länge mit einer Adresse und Betreffdaten im folgenden Format, das auf RFC 2368, dem `mailto`-URL-Schema basiert:

`benutzername@host?name1=wert1&name2=wert2...`

Dabei gilt: `benutzername@host` gibt einen Empfänger an. Die Name-Wert-Paare können in beliebiger Reihenfolge angegeben werden und werden wie folgt definiert:

- `from=benutzername@host` (darf nur einmal auftreten). Das `from`-Feld muss angegeben werden.
- `to=benutzername@host`. Verwenden Sie dieses Name-Wert-Paar für zusätzliche Empfänger.
- `reply-to=benutzername@host` (kann nur einmal vorhanden sein)
- `cc=benutzername@host`
- `bcc=benutzername@host`
- `subject=string` (kann nur einmal vorhanden sein)

Wenn die Felder `from`, `reply-to` oder `subject` öfter als einmal angegeben werden, wird die E-Mail-Nachricht nicht gesendet, die Funktion gibt den Wert `false()` zurück und der entsprechende Fehlerstatus wird gesetzt.

- Die Web Publishing Engine überprüft die Syntax aller angegebenen E-Mail-Adressen. Sie müssen die folgende Form haben:

`user@host.tld` oder "genannter Kennzeichner"<`user@host.tld`>

Dabei gilt: `tld` ist eine beliebige Domäne der oberen Ebene wie `com` oder `net`. Wenn beliebige dieser Felder eine ungültige E-Mail-Adresse enthalten, wird die E-Mail-Nachricht nicht gesendet und der entsprechende Fehlerstatuscode wird gesetzt.

- Die einzelnen Werte für den Parameter `smtpFelder`, z. B. der Betreff (`subject`), müssen URL-kodierte Strings sein. Beispiel: Das Zeichen `&` muss als `&` und Leerzeichen als `%20` angegeben werden. Der vollständige String für den Parameter `smtpFelder` muss XML-kodiert sein (siehe das Beispiel am Ende dieses Abschnitts).
- Für jede dieser Funktionen wird der Wert `true()` zurückgegeben, wenn die E-Mail-Nachricht erfolgreich gesendet wurde; anderenfalls wird `false()` zurückgegeben.
- Für E-Mail-Nachrichten verwendet die Web Publishing Engine anfangs die Textkodierung ISO-8859-1 als Standardeinstellung. Sie können diese Einstellung über Admin Console ändern. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

- Die Funktion `fmxml:send_email(String smtpFelder, String xsltDatei, Node xml, boolean inklBilder)` sendet eine E-Mail-Nachricht bestehend aus XML-Daten, die durch das in dieser Funktion angegebene E-Mail-Stylesheet verarbeitet werden.
 - Geben Sie für den Parameter `xsltDatei` den Namen des E-Mail-Stylesheets an, indem Sie einen URL eingeben, der relativ zur Hauptverarbeitungs-Stylesheet-Datei für die Abfrage ist.
 - Geben Sie für den Parameter `xml` den übergeordneten Knoten der XML-Daten an, die Sie mit dem E-Mail-Stylesheet verwenden wollen. Um eine E-Mail-Nachricht mit denselben XML-Daten zu senden, die im Browser angezeigt werden, geben Sie einfach den XPath für das Root-Verzeichnis des Dokuments an: `"/`. Anderenfalls können Sie ein anderes XML-Dokument verwenden, indem Sie es zunächst mithilfe der Funktion `document()` laden und es dann an die Funktion `fmxml:send_email()` übergeben.
 - Geben Sie für den Parameter `inklBilder` den booleschen Wert `true()` an, damit die Web Publishing Engine alle Bilder einschließt, die in der HTML der E-Mail-Nachricht als Anlagen angegeben sind. Dieser Parameter schließt sowohl die FileMaker-Datenbankbilder als auch andere Bilder von anderen Speicherorten ein. Die Web Publishing Engine ändert die Bild-URLs so, dass sie auf die Anlagen verweisen. Bei sehr zahlreichen oder großen Bildern kann die Leistung beeinträchtigt werden. Wenn Sie `false ()` angeben, ändert die Web Publishing Engine die URLs für die Bilder nicht. Wenn die URLs absolut sind, versucht der E-Mail-Client, die Bilder vom Web-Server zu laden.

Beispiel für die Verwendung der Funktion `fmxml:send_email(String smtpFelder, String xsltDatei, Node xml, boolean inklBilder)` in einer XPath-Anweisung, z. B. im `<xsl:if>`-Element:

```
fmxml:send_email('tom_jones@company.com?subject=project%20status&from=john_smith@company.com
&cc=jane_doe@company.com','my_mail_template.xml',/, true())
```

Informationen über die Konfiguration der Web Publishing Engine für die Verbindung zu einem SMTP-Server finden Sie in der FileMaker Server Hilfe.

Verwenden von Header-Funktionen

Sie können die Funktion `fmxml:get_header()` verwenden, um Informationen aus der HTTP-Abfrage und Antwort-Headern zu lesen, und die Funktion `fmxml:set_header()`, um Informationen in die Header zu schreiben. Diese Funktionen sind nützlich, wenn der Client mithilfe der Header-Informationen Daten aus dem Web-Server abrufen kann oder wenn Sie aus anderen Gründen einen HTTP-Header einrichten müssen.

Header-Erweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
<code>fmxml:get_header(String name)</code>	String	Gibt den angegebenen Header-Wert zurück.
<code>fmxml:set_header(String name, String wert)</code>	Ungültig	Legt den angegebenen Header-Wert fest.
<code>fmxml:set_status_code(Number status-code)</code>	Ungültig	Setzt den HTTP-Statuscode.

Hinweise

- Der in den Funktionen `fmxml:get_header()` und `fmxml:set_header()` verwendete Name sowie der in der Funktion `fmxml:set_header()` verwendete Wert können ein String beliebiger Länge sein.
- Die Funktion `fmxml:set_header()` und die Funktion `fmxml:set_status_code()` verlangen die Verarbeitungsanweisung `<?xslt-cwp-buffer?>`. Weitere Informationen finden Sie unter „Verwenden von Inhaltspufferung“ auf Seite 68.

Die folgenden Beispiele demonstrieren, wie der Wert des Headers gesetzt wird. Nehmen Sie an, Sie verwenden ein Stylesheet zur Ausgabe einer vCard. Wenn ein Browser versucht, die Stylesheet-Seite zu laden, besteht potenziell das Problem, dass der Browser die .xsl-Datei als Stylesheet anstatt als vCard interpretieren könnte. Wenn Sie den Header „Inhalt-Disposition“ verwenden, können Sie angeben, dass eine Anlage mit der Erweiterung .vcf vorhanden ist.

```
<xsl:value-of select="fmxslt:set_header('Inhalt-Disposition','attachment;filename=test.vcf')"/>
```

Verwenden der Cookie-Erweiterungsfunktionen

Mithilfe der Cookie-Erweiterungsfunktionen können Sie Cookies abrufen oder einrichten, die im Webbrowser des Clients gespeichert sind.

Cookie-Erweiterungsfunktion	Zurückgegebener Datentyp	Beschreibung
fmxslt:get_cookie(String name)	Knotengruppe	Gibt die COOKIE-Knotengruppe <code>node-set</code> zurück, die den angegebenen Cookie-Namen enthält.
fmxslt:get_cookies()	Knotengruppe	Gibt die COOKIE-Knotengruppe (<code>node-set</code>) mit allen vom Client gelieferten Cookies zurück.
fmxslt:set_cookie(String name, String wert)	Ungültig	Speichert das angegebene Cookie mit dem angegebenen Wert im Browser des Clients.
fmxslt:set_cookie(String name, String wert, Number ablauf, String pfad, String domäne)	Ungültig	Speichert das angegebene Cookie mit allen für ein Cookie verfügbaren Werten im Browser des Clients. Der Parameter <code>ablauf</code> gibt die Anzahl an Sekunden bis zum Ablauf des Cookies an.

Hinweise

- Die Funktionen `fmxslt:get_cookie()` und `fmxslt:get_cookies()` geben eine Knotengruppe (`node-set`) mit der folgenden Struktur zurück:


```
<!ELEMENT cookies (cookie)*>
  <!ATTLIST cookie xmlns CDATA #FIXED "http://www.filemaker.com/xml/cookie">
<!ELEMENT cookie (#PCDATA)>
  <!ATTLIST cookie name CDATA #REQUIRED>
```
- Der XML-Namespaces für die Cookies `node-set` ist `http://www.filemaker.com/xml/cookie`. Stellen Sie sicher, dass Sie den Namespace deklarieren und einen Präfix für den Namespace angeben.
- Alle Parameterwerte für die `fmxslt:set_cookie`-Funktionen müssen gültig sein, da der Webbrowser sonst die `fmxslt:set_cookie`-Funktionsabfragen ignoriert.
- Für alle Cookie-Funktionen können die String-Parameter eine beliebige Länge aufweisen.
- Beide Formen der Funktion `fmxslt:set_cookie()` verlangen die Verarbeitungsanweisung `<?xslt-cwp-buffer?>`. Weitere Informationen finden Sie unter „Verwenden von Inhaltspufferung“ auf Seite 68.

Beispiel: get_cookie

Das folgende Beispiel ruft einen Cookie `preferences` und seinen Wert ab:

```
<xsl:variable name="pref_cookie" select="fmxslt:get_cookie('preferences')"/>
<xsl:value-of select="concat('Cookie Name = ', $pref_cookie/fmc:cookies/fmc:cookie/@name)"/> <br/>
<xsl:value-of select="concat('Cookie Value = ', $pref_cookie/fmc:cookies/fmc:cookie)"/>
```

Beispiel: set_cookie

Beispiel für das Setzen eines Cookies mit allen Werten:

```
<xsl:variable name="storing_cookie" select="fmxslt:set_cookie ('text1', 'text2', 1800, 'my_text', 'my.company.com')"/>
```

Verwenden der Erweiterungsfunktionen zur Stringmanipulation

Sie können mithilfe der Funktionen zur Stringmanipulation die Kodierung von Strings beliebiger Länge ändern.

Erweiterungsfunktion zur Stringmanipulation	Zurückgegebener Datentyp	Beschreibung
fmxslt:break_encode(String wert)	String	Gibt einen HTML-Break-kodierten String zurück. Zeichen wie & (kaufmännisches UND) werden durch & ersetzt. Zeilenumbruchzeichen wie Zeilenvorschub und Absatzende-Zeichen werden durch ersetzt. Diese Funktion ist nur wirksam, wenn das Attribut disable-output-escaping der Elemente <xsl:value-of> und <xsl:text> auf „yes“ gesetzt ist (disable-output-escaping="yes"). Hinweis Um ein Zeilenvorschub- oder Absatzende-Zeichen in den String einzufügen, auf den die Funktion fmxslt:break_encode() angewendet wird, müssen Sie die folgenden Escape-Zeichen im String verwenden: “
 (für Zeilenvorschub) bzw.  (für Absatzende). Sie können kein Zeilenvorschub- oder Absatzende-Zeichen in den String einfügen, indem Sie in Ihrem Texteditor den Zeilenschalter drücken.
fmxslt:html_encode(String wert)	String	Gibt einen HTML-kodierten String zurück; Zeichen wie & (kaufmännisches UND) werden durch & ersetzt.
fmxslt:url_encode(String wert)	String	Gibt einen URL-kodierten String zurück. URL-Kodierung wird insbesondere für URLs verwendet, um Zeichen über das Internet zu übertragen. Zum Beispiel ist %26 die URL-kodierte Form von & (kaufmännisches Und). Wenn ein reserviertes Zeichen in Ihrem href verwendet wird, benutzen Sie diese Funktion zur URL-Kodierung Ihres Strings.
fmxslt:url_encode(String wert, String kodierung)	String	Gibt einen URL-kodierten String mit der Zeichenkodierung zurück, die Sie im Parameter kodierung angeben. Mögliche Kodierungen sind: US-ASCII, ISO-8859-1, ISO-8859-15, ISO-2022-JP, Shift_JIS und UTF-8. Verwenden Sie diese Funktion in Situationen, in denen Sie wissen, dass sich die von einem Web-Server erwartete Zeichenkodierung von derjenigen unterscheidet, die in Ihrem aktuellen Stylesheet verwendet wird. Beispiel: Ihre Website-Einstiegsseite kann in UTF-8 angezeigt sein, aber Benutzer klicken auf einen Link, um zu einer japanischen Seite zu wechseln. Wenn die Abfrage japanische Zeichen enthält und die japanischen Seiten Shift_JIS-Kodierung verwenden, wird der String am besten in Shift_JIS kodiert.

Erweiterungsfunktion zur Stringmanipulation	Zurückgegebener Datentyp	Beschreibung
<code>fmxml:url_decode(String wert)</code>	String	Gibt einen URL-dekodierten String zurück, der zuvor kodiert war.
<code>fmxml:url_decode(String wert, String kodierung)</code>	String	Gibt einen URL-dekodierten String mit der Zeichenkodierung zurück, die Sie im Parameter <code>kodierung</code> angeben. Mögliche Kodierungen sind: US-ASCII, ISO-8859-1, ISO-8859-15, ISO-2022-JP, Shift_JIS und UTF-8. Verwenden Sie diese Funktion in Situationen, in denen Sie die in einem URL-kodierten String verwendete Zeichenkodierung angeben müssen, um den String korrekt zu dekodieren. Beispiel: Selbst wenn Ihre Website ISO-8859-1 verwendet, könnten Benutzer ein Formular mit einer anderen Zeichenkodierung senden.

Vergleich von Strings mithilfe regulärer Perl 5-Ausdrücke

Sie können die Erweiterungsfunktion `fmxml:regex_contains()` verwenden, um Strings mithilfe regulärer Perl 5-Ausdrücke zu vergleichen. Ein Vergleich mit regulären Ausdrücken ist eine fortschrittliche Art von Textabgleich, bei der Sie bestimmen können, ob ein String einem bestimmten Muster entspricht. Syntax dieser Funktion:

```
fmxml:regex_contains(String eingabe, String muster)
```

Dabei gilt: `eingabe` ist ein String und `muster` ist ein regulärer Perl 5-Ausdruck. Weitere Informationen zur Syntax von regelmäßigen Perl 5-Ausdrücken finden Sie unter www.perldoc.com. Die Funktion `fmxml:regex_contains()` gibt einen booleschen Wert zurück.

Diese Funktion ist nützlich, wenn Sie anspruchsvollere Möglichkeiten zur Stringmanipulation benötigen, die von Standard-XSLT nicht zur Verfügung gestellt werden. Sie können z. B. feststellen, ob ein Feldwert eine gültige Telefonnummer oder E-Mail-Adresse enthält, indem Sie den String mit einem regulären Perl 5-Ausdruck vergleichen.

Beispiel für den Einsatz dieser Funktion, um zu ermitteln, ob ein Feldwert korrekt konstruierte E-Mail-Adressen enthält:

```
<xsl:variable name="email" select="'foo@bar.com'"/>
<xsl:if test="fmxml:regex_contains($email, '^\\w+([\\w-\\.]*@\\w+((-[\\w+])|(\\w*))\\. [a-z]{2,3}$)')">Valid Email</xsl:if>
```

Wenn die Web Publishing Engine das Muster nicht analysieren kann, wird der Fehlerstatus auf Fehlercode 10311 gesetzt. Weitere Informationen finden Sie unter „Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen“ auf Seite 121.

Prüfen von Werten in einem Feld, das als Kontrollkästchen formatiert ist

Mithilfe der folgenden Erweiterungsfunktion können Sie ermitteln, ob ein bestimmter Wert in einer Kontrollkästchen-Werteliste in einem Feld in der FileMaker-Datenbank gespeichert ist:

```
fmxml:contains_checkbox_value(String werteString, String werteListenEintrag)
```

Dabei gilt: `werteString` ist ein XPath, der das Feld angibt, und `werteListenEintrag` ist der Wert, den Sie suchen.

Wenn der angegebene Wert im Feld gespeichert ist, gibt diese boolesche Funktion `true()` zurück. Anderenfalls gibt sie `false()` zurück. Sie können anhand dieser Funktion ermitteln, ob das geprüfte Attribut in einem HTML-Formular so eingestellt werden soll, dass es ein Kontrollkästchen als markiert zeigt.

Nehmen Sie z. B. an, ein Feld in einem FileMaker-Datenbanklayout hat die folgenden Kontrollkästchenoptionen:

- Rot
- Blau
- Grün
- Klein
- Mittel
- Groß

Wenn ein Benutzer nur „Rot“ auswählt, würde das Feld den String Rot enthalten. Anhand des folgenden Funktionsaufrufs könnten Sie feststellen, ob das Feld Blau enthält:

```
fmxml:contains_checkbox_value(<feldwert_knoten>,'Blau')
```

Dabei gilt: <feldwert_knoten> ist der XPath zum <data>-Element für das Kontrollkästchenfeld. Die Funktion würde in diesem Beispiel false zurückgeben.

Eine übliche Anwendung dieser Funktion ist die Anzeige der Kontrollkästchen-Werteliste in einer Web-Seite und die Auswahl der Kontrollkästchen auf der Web-Seite, die auch in der Datenbank ausgewählt sind. Die folgenden HTML- und XSLT-Anweisungen erstellen eine Gruppe von Kontrollkästchen für ein Feld mit dem Namen style, das eine Werteliste mit dem Namen color_size benutzt:

```
<xsl:variable name="field-value" select="fmrs:field[@name='style']/fmrs:data" />
<xsl:for-each select="$valuelists[@NAME = 'color_size']/fml:VALUE">
  <input type="checkbox">
    <xsl:attribute name="name">style</xsl:attribute>
    <xsl:attribute name="value"><xsl:value-of select="."/></xsl:attribute>
    <xsl:if test="fmxml:contains_checkbox_value($field-value,.)">
      <xsl:attribute name="checked">checked</xsl:attribute>
    </xsl:if>
  </input><xsl:value-of select="."/><br/>
</xsl:for-each>
```

Die HTML- und XSLT-Anweisungen im Beispiel würden die folgenden Kontrollkästchen auf einer Web-Seite ausgeben, von denen Rot und Mittel ausgewählt sind:

- Rot
- Blau
- Grün
- Klein
- Mittel
- Groß

Verwenden der Datums-, Zeit- und Tages-Erweiterungsfunktionen

Mithilfe von Erweiterungsfunktionen können Sie die aktuellen Werte für Datum, Uhrzeit oder Tag abrufen und zwei Datums-, Zeit- oder Tagesangaben vergleichen.

Die Funktionen in der folgenden Tabelle verwenden die „fm“-Formate unabhängig von den regionalen Einstellungen. Die „fm“-Formate sind MM/tt/jjjj für Datum, HH:mm:ss für Zeit und MM/tt/jjjj HH:mm:ss für Zeitstempel.

Um ein anderes oder bevorzugtes Format für die Ausgabewerte zu verwenden, verwenden Sie Berechnungsfunktionen oder JavaScript.

Datums-, Zeit-, Tages-Erweiterungsfunktionen	Zurückgegebener Datentyp	Beschreibung
fmxls:get_date()	String	Gibt das aktuelle Datum im „fm“-Format zurück.
fmxls:get_date(String format)	String	Gibt das aktuelle Datum in dem Format zurück, das Sie angeben. Geben Sie den String „short“ oder „long“ für den Parameter format ein.
fmxls:get_time()	String	Gibt die aktuelle Zeit im „fm“-Format zurück.
fmxls:get_time(String format)	String	Gibt die aktuelle Uhrzeit in dem Format zurück, das Sie angeben. Geben Sie den String „short“ oder „long“ für den Parameter format ein.
fmxls:get_day()	String	Gibt den aktuellen Tag im Kurzformat zurück.
fmxls:get_day(String format)	String	Gibt den aktuellen Tag in dem Format zurück, das Sie angeben. Geben Sie den String „short“ oder „long“ für den Parameter format ein.
fmxls:get_fm_date_format()	String	Gibt den Formatierungsstring für das „fm“-Datumsformat zurück: „MM.tt.jjjj“
fmxls:get_short_date_format()	String	Gibt den Formatierungsstring für das kurze Datumsformat zurück: „M/t/jj“
fmxls:get_long_date_format()	String	Gibt den Formatierungsstring für das lange Datumsformat zurück: „MMM t, jjj“
fmxls:get_fm_time_format()	String	Gibt den Formatierungsstring für das „fm“-Zeitformat zurück: „HH:mm:ss“
fmxls:get_fm_timestamp_format()	String	Gibt den Formatierungsstring für das „fm“-Zeitstempelformat zurück: „MM/tt/jjjj HH:mm:ss“
fmxls:get_short_time_format()	String	Gibt den Formatierungsstring für das kurze Zeitformat zurück: „hh:mm a“
fmxls:get_long_time_format()	String	Gibt den Formatierungsstring für das lange Zeitformat zurück: „hh:mm:ss a z“
fmxls:get_short_day_format()	String	Gibt den Formatierungsstring für das kurze Tagesformat zurück: „EEE“
fmxls:get_long_day_format()	String	Gibt den Formatierungsstring für das lange Tagesformat zurück: „EEEE“
fmxls:compare_date(String datum1, String datum2)	Zahl	Diese Funktion vergleicht zwei Datumswerte. Sie gibt eine negative Zahl zurück, wenn datum1 vor datum2 liegt. Sie gibt eine positive Zahl zurück, wenn datum1 nach datum2 liegt. Sie gibt eine 0 zurück, wenn datum1 identisch mit datum2 ist. Beide Datumswerte müssen im „fm“-Format angegeben sein.

Datums-, Zeit-, Tages-Erweiterungsfunktionen	Zurückgegebener Datentyp	Beschreibung
fmxls:compare_time(String zeit1, String zeit2)	Zahl	Diese Funktion vergleicht zwei Zeitwerte. Sie gibt eine negative Zahl zurück, wenn zeit1 vor zeit2 liegt. Sie gibt eine positive Zahl zurück, wenn zeit1 nach zeit2 liegt. Sie gibt eine 0 zurück, wenn zeit1 identisch mit zeit2 ist. Beide Zeitwerte müssen im „fm“-Format angegeben sein.
fmxls:compare_day(String tag1, String tag2)	Zahl	Diese Funktion vergleicht zwei Tageswerte. Sie gibt eine negative Zahl zurück, wenn tag1 vor tag2 liegt. Sie gibt eine positive Zahl zurück, wenn tag1 nach tag2 liegt. Sie gibt eine 0 zurück, wenn tag1 identisch mit tag2 ist. Beide Tage müssen im kurzen Tagesformat angegeben sein.

Die Funktionen in der folgenden Tabelle verwenden spezielle Formatstrings, die ein Datums- und Zeitformat angeben. Weitere Informationen finden Sie im nächsten Abschnitt, „Erläuterung von Datums- und Zeitformatstrings“.

Datums-, Zeit-, Tages-Erweiterungsfunktionen	Zurückgegebener Datentyp	Beschreibung
fmxls:get_datetime(String datumsFormat)	String	Gibt das aktuelle Datum und die Uhrzeit zurück und verwendet dafür die Datums- und Zeitformatstrings.
fmxls:convert_datetime(String altesFormat, String neuesFormat, String datum)	String	Konvertiert das angegebene Datum aus altesFormat in einen String in der Form von neuesFormat. Die Strings altesFormat und neuesFormat müssen mithilfe der Datums- und Zeitformatstrings angegeben werden.
fmxls:compare_datetime(String datumsFormat1, String datumsFormat2, String datum1, String datum2)	Zahl	Diese Funktion vergleicht datum1 und datum2, indem sie die Datumsangaben gemäß den jeweiligen Datumsformaten dekodiert. Sie gibt eine negative Zahl zurück, wenn datum1 vor datum2 liegt. Sie gibt eine positive Zahl zurück, wenn datum1 nach datum2 liegt. Sie gibt eine 0 zurück, wenn datum1 identisch mit datum2 ist. Die Strings datumsFormat1 und datumsFormat2 müssen mithilfe der Datums- und Zeitformatstrings angegeben werden.

Erläuterung von Datums- und Zeitformatstrings

Die Datums- und Zeitformate werden durch Datums- und Zeitmusterstrings angegeben. Innerhalb von Datums- und Zeitmusterstrings werden Buchstaben ohne Anführungszeichen von A bis Z und von a bis z als Musterbuchstaben interpretiert, die Komponenten des Datums- oder Zeitstrings darstellen.

Die folgenden Musterbuchstaben sind definiert (alle anderen Zeichen von A bis Z und a bis z sind reserviert):

Buchstabe	Datums- oder Zeitkomponente	Präsentation	Beispiele
G	Ärabezeichnung	Text	n. Chr.
j	Jahr	Jahr	1996; 96
M	Monat im Jahr	Monat	Juli; Jul; 07
w	Woche im Jahr	Zahl	27
W	Woche im Monat	Zahl	2
B	Tag im Jahr	Zahl	189
t	Tag im Monat	Zahl	10

Buchstabe	Datums- oder Zeitkomponente	Präsentation	Beispiele
F	Tag der Woche im Monat	Zahl	2
U	Wochentag	Text	Tuesday, Tue
a	AM/PM-Markierung	Zahl	PM
H	Stunde des Tages (0-23)	Zahl	0
k	Stunde des Tages (1-24)	Zahl	24
K	Stunde vormittags/nachmittags (0-11)	Zahl	0
h	Stunde vormittags/nachmittags (1-12)	Zahl	12
m	Minute in Stunde	Zahl	30
s	Sekunde in Minute	Zahl	55
S	Millisekunde	Zahl	978
z	Zeitzone	Allgemeine Zeitzone	Pazifische Sommerzeit; PDT; GMT-08:00
Z	Zeitzone	RFC 822 Zeitzone	-0800

Musterbuchstaben werden gewöhnlich wiederholt, da ihre Anzahl die exakte Darstellung bestimmt:

- **Text:** Wenn beim Formatieren vier oder mehr Musterbuchstaben vorhanden sind, wird die vollständige Form verwendet. Anderenfalls wird falls verfügbar eine kurze oder abgekürzte Form verwendet. Beim Parsing werden beide Formen unabhängig von der Anzahl der Musterbuchstaben berücksichtigt.
- **Zahl:** Beim Formatieren gibt die Anzahl der Musterbuchstaben die minimale Stellenanzahl an, Zahlen mit weniger Stellen werden bis zu dieser Anzahl mit Nullen aufgefüllt. Beim Parsing wird die Anzahl der Musterbuchstaben ignoriert, sofern sie nicht zur Trennung zweier benachbarter Felder erforderlich sind.
- **Jahr:** Wenn beim Formatieren zwei Musterbuchstaben vorhanden sind, wird das Jahr auf zwei Stellen abgeschnitten. Anderenfalls wird es als eine Zahl interpretiert.

Wenn beim Parsing mehr als zwei Musterbuchstaben vorhanden sind, wird das Jahr unabhängig von der Stellenanzahl buchstäblich interpretiert. Bei Verwenden des Musters MM/tt/jjjj wird die Angabe 01/11/12 als 11. Jan 12 A.D. analysiert.

Beim Parsing mit dem abgekürzten Jahresmuster (y oder yy) muss das abgekürzte Jahr in Relation zu einem Jahrhundert interpretiert werden, indem Datumsangaben so angepasst werden, dass sie im Zeitraum von 80 Jahren vor und 20 Jahren nach der Erstellung der Instanz liegen. Wenn Sie z. B. das Muster MM/tt/jj verwenden und eine Instanz mit diesem Datumsformat am 1. Januar 1997 erstellt wurde, wird der String „11.01.12“ als 11. Januar 2012 interpretiert. Dagegen wird der String „05/04/64“ als 4. Mai 1964 interpretiert. Beim Parsing werden nur Strings mit exakt zwei Stellen als das Standardjahrhundert analysiert. Jeder andere numerische String, z. B. ein einstelliger String, ein drei- oder mehrstelliger String oder ein zweistelliger String, der nicht ausschließlich aus Ziffern besteht (z. B. -1), wird buchstäblich interpretiert. Also werden 01.02.3 oder 01.02.003 mit demselben Muster wie 2. Jan 3 AD analysiert. Ebenso wird 01.02.-3 als 2. Jan 4 v. Chr. interpretiert.

- **Monat:** Wenn zwei oder mehr Musterbuchstaben vorhanden sind, wird der Monat als Text interpretiert, anderenfalls als Zahl.

- **Allgemeine Zeitzone:** Zeitzonen werden als Text interpretiert, wenn sie Namen haben. Für Zeitzonen, die einen von GMT versetzten Wert haben, wird folgende Syntax verwendet:
 - GMTVersatzZeitzone. *GMT Vorzeichen Stunden:Minuten*
 - Vorzeichen. + oder -
 - Stunden. *Ziffer* oder *Ziffer Ziffer*
 - Minuten. *Ziffer Ziffer*
 - Ziffer. Eines von: 0 1 2 3 4 5 6 7 8 9

Stunden müssen zwischen 0 und 23 liegen, *Minuten* zwischen 00 und 59. Das Format ist unabhängig von der Region und die Ziffern müssen dem Basic-Latin-Block des Unicode-Standards entnommen werden.

Beim Parsing werden auch RFC 822-Zeitzone akzeptiert.

- **RFC 822-Zeitzone:** Beim Formatieren wird das vierstellige RFC 822-Zeitzoneformat verwendet:
 - RFC822Zeitzone. *Vorzeichen ZweiStelligeStunden Minuten*
 - ZweiStelligeStunden. *Ziffer Ziffer*

ZweiStelligeStunden muss zwischen 00 und 23 liegen. Andere Definitionen sind wie für allgemeine Zeitzone.

Beim Parsing werden auch allgemeine Zeitzone akzeptiert.

Die folgenden Beispiele zeigen, wie Datums- und Zeitmuster in den USA interpretiert werden. Die angegebenen Datums- und Zeitwerte sind 2001-07-04 12:08:56 Ortszeit in der US-amerikanischen Zeitzone „Pacific Time“.

Datums- und Zeitmuster	Ergebnis
"jjj.MM.tt G 'at' HH:mm:ss z"	2001.07.04 n. Chr. at 12:08:56 PDT
"EEE, MMM t, 'jj"	Mi, Jul 4, '01
"hh:mm"	12:08 PM
"hh 'o' 'clock' a, zzzz"	12 o'clock PM, Pazifische Sommerzeit
"K:mm a, z"	0:08 PM, PDT
"jjj.MMMMMM.tt GGG hh:mm aaa"	02001.Juli.04 n. Chr. 12:08 PM
"EEE, t MMM jjj HH:mm:ss Z"	Mi, 4 Jul 2001 12:08:56 -0700
"jjMMtHHmssZ"	010704120856-0700

Copyright 2003 Sun Microsystems, Inc. Genehmigter Nachdruck.

Prüfen des Fehlerstatus von Erweiterungsfunktionen

Sie können mithilfe der folgenden Erweiterungsfunktion in einem XSLT-Stylesheet den aktuellen Fehlerstatus der zuletzt aufgerufenen FileMaker XSLT-Erweiterungsfunktion prüfen und Fehler behandeln, die bei der Verarbeitung Ihrer Seiten auftreten:

```
fmxslt:check_error_status()
```

Beim Aufruf der Funktion `fmxslt:check_error_status()` gibt die Web Publishing Engine den aktuellen Fehlercode für die zuletzt aufgerufene Funktion im Typ Zahl zurück und setzt den Fehlerstatus dann auf 0 („Kein Fehler“) zurück. Informationen über die Fehlercodewerte finden Sie unter „Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen“ auf Seite 121.

Verwenden der Protokollierung

Sie können mithilfe des XSLT-Standardelements `<xsl:message>` Protokolleinträge in die Protokolldatei der Web Publishing Engine schreiben. Weitere Informationen finden Sie unter „Verwenden des Web Publishing Engine-Programmprotokolls“ auf Seite 90.

Verwenden der Server-seitigen Verarbeitung von Scriptingsprachen

Der zugrunde liegende XSLT-Transformer, der in die Web Publishing Engine integriert ist, unterstützt die Server-seitige Verarbeitung von Scriptingsprachen. Daher können Sie JavaScript verwenden, um Ihre eigenen Erweiterungsfunktionen zu entwickeln, die direkt aus einem XSLT-Stylesheet aufgerufen werden können.

Zwei Java-Libraries werden installiert, um diese Funktionalität zu ermöglichen:

- `bsf.jar` – Diese Library ermöglicht dem XSLT-Transformer, Verbindungen zu Scriptingsprachen herzustellen.
- `js.jsr` – Diese Library ist eine vollständige JavaScript-Implementierung des Mozilla-Projekts.

Mit diesen Libraries können Sie Ihre eigenen Erweiterungsfunktionen innerhalb Ihres XSLT-Stylesheet-Codes erstellen. Diese Erweiterungsfunktionen können jede Scriptinglogik implementieren und sind besser zu verwalten, als XSLT und XPath zu verwenden, um logische Funktionen umzusetzen.

Weitere Informationen über die Erweiterungsunterstützung des XSLT-Transformers finden Sie auf der Apache Xalan Extensions-Website:

<http://xml.apache.org/xalan-j/extensions.html>

Definieren einer Erweiterungsfunktion

So definieren Sie eine Erweiterungsfunktion innerhalb Ihres Stylesheets:

1. Definieren Sie den Namespace für die Erweiterung.

Fügen Sie den Namespace `xalan` hinzu, um den XSLT-Transformer anzuweisen, Erweiterungskomponenten zu unterstützen, indem Sie den Namen Ihres eigenen Erweiterungsfunktion-Namespace angeben. Das folgende Beispiel verwendet `fmp-ex` als Namespace-Präfix der Erweiterungsfunktion.

```
<xsl:stylesheet version="1.0"
xmlns:xsl=http://www.w3.org/1999/XSL/Transform
xmlns:xalan=http://xml.apache.org/xslt
xmlns:fmp-ex="ext1"
exclude-result-prefixes="xsl xalan fmp-ex">
```

2. Definieren Sie die Erweiterungskomponente und die Erweiterungsfunktionen mit dem Code, der tatsächlich Ihre Erweiterungsfunktion implementiert.

```
<xalan:component prefix="fmp-ex" functions="getValueColor">
<xalan:script lang="javascript">
  function getValueColor(value) {
    if (value > 0)
      return ("#009900");
    else
      return ("#CC0000");
  }
</xalan:script>
</xalan:component>
```

Dieses Beispiel gibt einen Farbwert basierend auf einem Eingabewert zurück. Wenn der Eingabewert größer als 0 ist, ist die zurückgegebene Farbe Grün ("009900"). Wenn anderenfalls der Wert kleiner 0 ist, ist die zurückgegebene Farbe Rot ("CC0000").

Hinweis Das Element `<xalan:component>` muss dem Element `<xsl:stylesheet>` untergeordnet sein.

3. Verwenden Sie die Erweiterungsfunktion innerhalb des Stylesheets.

Das folgende Beispiel zeigt, wie Sie eine Erweiterungsfunktion mittels einer XPath-Anweisung aufrufen.

Das erste Beispiel würde die Schriftfarbe auf Grün ("009900") setzen.

```
<font color="{fmp-ex:getValueColor(50)}">Der Wert ist 50</font>
```

Das zweite Beispiel würde die Schriftfarbe auf Rot ("CC0000") setzen.

```
<font color="{fmp-ex:getValueColor(-500)}">Der Wert ist -500</font>
```

Ein Beispiel für eine Erweiterungsfunktion

Die einfache JavaScript-Funktion, die im obigen Prozess verwendet wird, könnte über eine `<xsl:choose>`-Anweisung implementiert worden sein. Aber der Kniff bei der Verwendung einer Scriptingerweiterung liegt darin, dass Sie eine Funktion erstellen können, die nicht alleine in XSLT oder XPath implementiert werden kann.

Nehmen Sie zum Beispiel an, dass Sie ein Intranet-Portal für Ihr Unternehmen aufsetzen und die aktuellen Aktiennotierungen auf dieser Portalseite aufnehmen möchten. Es gibt zwar XML-Börsen-Feeds, im Allgemeinen sind jedoch kommerzielle Lizenzen erforderlich, um auf sie zuzugreifen. Sie können jedoch Börsendaten in einem CSV-Dokument von der Yahoo!-Website herunterladen. Die XPath document()-Funktion kann Inhalte aus XML-Quellen importieren, aber Sie müssten den CSV-Inhalt zu XML konvertieren. Eine Lösung besteht darin, dass Sie JavaScript verwenden, um die CSV-Börsendaten herunterzuladen, die Datei zu parsen und die Daten zu extrahieren.

Dieser URL zeigt die Syntax für den Abruf einer Börsennotierung von der Yahoo!-Website in Form einer CSV-Datei:

```
http://quote.yahoo.com/d/quotes.csv?s=<ticker>&f=l1gh&e=.csv
```

Dabei gilt: <ticker> steht für das Börsensymbol, für das Sie Daten abrufen möchten.

Die zurückgegebenen Daten bestehen aus drei Zahlen, die durch Kommata getrennt sind. Beispiel:

```
31.79,31.17,32.12
```

Der erste Wert ist der letzte Handelspreis, der zweite Wert das Tagesstief und der dritte Wert das Tageshoch.

Das folgende Beispiel zeigt eine JavaScript-XSLT-Erweiterungsfunktion, die einen aktuellen Börsenkurs von der Yahoo!-Website abrufen und ihn über eine XPath-Funktion zur Verfügung stellt:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  exclude-result-prefixes="xsl fmxslt fmrs xalan fmp-ex"
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset"
  xmlns:fmxslt="xalan://com.fmi.xslt.ExtensionFunctions"
  xmlns:xalan="http://xml.apache.org/xslt"
  xmlns:fmp-ex="ext1"
>

<?xslt-cwp-query params="-grammar=fmresultset&-process" ?>
<xsl:output method="html"/>
<xalan:component prefix="fmp-ex" functions="include get_quote" >
<xalan:script lang="javascript">
  function include(url) {
    var dest = new java.net.URL(url);
    var dis = new java.io.DataInputStream(dest.openStream());
    var res = "";
    while ((line = dis.readLine()) != null)
    {
      res += line + java.lang.System.getProperty("line.separator");
    }
    dis.close();
    return res;
  }
  function get_quote(ticker) {
    url = "http://quote.yahoo.com/d/quotes.csv?s="+
```

```

        "+ticker+"&f=l1gh&e=.csv";
    csv_file = include(url);
    var str_tokenizer = new java.util.StringTokenizer(csv_file, ',');
    // Der erste Token ist der letzte Handelspreis.
    var last = str_tokenizer.nextToken();
    return last;
}
</xalan:script>
</xalan:component>

<xsl:template match="/fmrs:fmresultset">
    <html>
        <Datenbereich>
        <font size="2" face="verdana, arial">
        Apple Stock Price: <xsl:value-of select="fmp-ex:get_quote('AAPL')"/>
        </font>
        </body>
    </html>
</xsl:template>

</xsl:stylesheet>

```

Wenn die Web Publishing Engine dieses Stylesheet verarbeitet, ruft es die Börsendaten von der Yahoo!-Website ab. Die Funktion `get_quote()` geht durch die Börsenkursdaten und gibt die Daten an das Stylesheet zurück. Die transformierte Ausgabe wird im Browser angezeigt.

Kapitel 7

Einstellen, Testen und Überwachen einer Site

Dieses Kapitel enthält Informationen über das Einstellen und Testen einer Custom Web Publishing-Site vor ihrem Einsatz in einer Produktionsumgebung. Zudem finden Sie hier Informationen zur Verwendung von Protokolldateien, um die Site während der Tests und nach der Freigabe zu überwachen.

Einstellen einer Custom Web Publishing-Site

Bevor Sie Ihre Site richtig testen können, müssen Sie die erforderlichen Dateien an die richtigen Speicherorte auf dem bzw. den Servern kopieren oder verschieben.

So stellen Sie Ihre Site zum Testen ein und bereiten Sie dafür vor:

1. Folgen Sie den Anleitungen in Kapitel 3, „Vorbereiten von Datenbanken für Custom Web Publishing“.
2. Stellen Sie sicher, dass XSLT und XML in FileMaker Server Admin Console aktiviert und richtig konfiguriert wurden.

Hinweis Weitere Informationen finden Sie in der FileMaker Server Hilfe.

3. Stellen Sie sicher, dass der Web-Server und die Web Publishing Engine laufen.
4. Kopieren oder verschieben Sie die XSLT-Stylesheets auf den Rechner, auf dem sich die Web Publishing Engine befindet.

Kopieren oder verschieben Sie die XSLT-Stylesheets in das folgende Verzeichnis auf dem Web Publishing Engine-Rechner:

- Apache (Mac OS): /Library/FileMaker Server/Web Publishing/xslt-template-files
- IIS (Windows): <Laufwerk>\Programme\FileMaker\FileMaker Server\Web Publishing\xslt-template-files

wobei <Laufwerk> das Hauptlaufwerk ist, von dem das System startet.

Hinweis Sie können die Stylesheets auch in einem optionalen Ordner oder einer Ordnerhierarchie innerhalb des Ordners xslt-template-files platzieren.

5. Kopieren oder verschieben Sie referenzierte Medienobjekte auf den Web-Server-Rechner.

Wenn ein Datenbank-Medienfeld einen Dateiverweis anstelle der eigentlichen Datei speichert, muss das referenzierte Medienobjekt im FileMaker Pro-Web-Ordner gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird. Um Ihre Site einzustellen, kopieren oder verschieben Sie die referenzierten Medien in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software.

Hinweis Falls die Medienfelder die tatsächlichen Dateien in der FileMaker-Datenbank speichern, ist mit dem Inhalt der Medienfelder keine Aktion erforderlich, wenn die Datenbank richtig auf der Datenbank-Server-Komponente des FileMaker Server-Einsatzes bereitgestellt und zugänglich ist.

6. Verwenden Sie folgende URL-Syntax, um ein XSLT-Stylesheet anzufordern und zu verarbeiten und das resultierende HTML zu generieren.

```
<Schema>://<Host>[:<Port>]/fmi/xsl/<Pfad>/<Stylesheet>.xsl[?<Query-String>]
```

Dabei gilt:

- <Schema> ist das HTTP- oder HTTPS-Protokoll.
- <Host> ist die IP-Adresse oder der Domänenname des Host-Computers, auf dem der Web-Server installiert ist.
- <port> optional ist und den Port angibt, den der Web-Server überwacht. Wird kein Port angegeben, wird der Standard-Port des Protokolls (Port 80 für HTTP oder Port 443 für HTTPS) angenommen.
- <path> ist optional und gibt den/die Ordner innerhalb des Ordners xslt-template-files an, in dem sich das XSLT-Stylesheet befindet.
- <Stylesheet> ist der Name des Stylesheets mit einer .xsl-Erweiterung.
- <query_string> kann eine Kombination aus einem Query-Befehl und einem oder mehreren Query-Parametern für Custom Web Publishing mit XSLT sein.

Wenn das angegebene Stylesheet eine <?xslt-cwp-query ?>-Verarbeitungsanweisung enthält, haben der statisch definierte Query-Befehl und die Parameter Vorrang vor anderen Query-Befehlen oder Parametern in der URL-Query-Zeichenfolge. Wenn Sie auf das Stylesheet home.xsl verweisen, das durch den XSLT-Site-Assistenten generiert wurde, müssen Sie keinen Query-String integrieren. Weitere Informationen zu Query-Strings finden Sie in Anhang A, „Gültige Namen in Query-Strings“.

Die URL-Syntax einschließlich der Namen der Query-Befehle und -Parameter unterscheidet mit Ausnahme von Teilen der Query-Zeichenfolge zwischen Groß- und Kleinschreibung. Der Großteil des URLs wird in Kleinbuchstaben angegeben. Wenn Sie z. B. Ihre Stylesheets (einschließlich eines home.xsl-Stylesheets) in den Ordner my_templates im Ordner xslt-template-files kopiert haben, können Sie folgenden URL verwenden, um die Stylesheets anzufordern und zu verarbeiten:

```
http://192.168.123.101/fmi/xsl/my_templates/home.xsl
```

Hinweis Die Web Publishing Engine erlaubt Web-Benutzern nicht, den Quelltext der XSLT-Stylesheets anzuzeigen, die im Ordner xslt-template-files installiert sind. Wenn ein Web-Benutzer eine Anforderung für die Verarbeitung eines Stylesheets sendet, sendet die Web Publishing Engine nur die HTML-Seiten an den Webbrowser, die das Ergebnis der XSLT-Site-Assistent-Stylesheets sind.

Testen einer Custom Web Publishing-Site

Bevor Sie Benutzern mitteilen, dass Ihre Custom Web Publishing-Site verfügbar ist, sollten Sie überprüfen, ob Aussehen und Funktionsweise Ihren Vorstellungen entsprechen.

- Testen Sie Funktionen wie das Suchen, Hinzufügen, Löschen und Sortieren von Datensätzen mit unterschiedlichen Konten und Zugriffsrechten.
- Überprüfen Sie, ob sich die Berechtigungen wie erwartet verhalten, indem Sie sich mit verschiedenen Konten anmelden. Stellen Sie sicher, dass nicht autorisierte Benutzer nicht auf Ihre Daten zugreifen oder sie ändern können.
- Überprüfen Sie alle Scripts, um sicherzustellen, dass das Ergebnis wie erwartet ist (siehe „FileMaker-Scripts und Custom Web Publishing“ auf Seite 24 für Informationen zur Gestaltung Web-freundlicher Scripts).
- Testen Sie Ihre Site mit unterschiedlichen Betriebssystemen und Webbrowsern.

Hinweis Wenn Sie über keine Netzwerkverbindung verfügen und FileMaker Server – also Web-Server, Web Publishing Engine und Datenbank-Server – auf einem einzigen Computer installiert haben, können Sie Ihre Custom Web Publishing-Site testen, indem Sie im URL Folgendes verwenden: `http://localhost/` oder `http://127.0.0.1/`. Informationen über die URL-Syntax finden Sie unter „Erläuterung der URL-Syntax für XML-Daten und Medienobjekte“ auf Seite 38, und „Erläuterung der URL-Syntax für FileMaker XSLT-Stylesheets“ auf Seite 56.

Beispiele für Stylesheets zum Testen der XML-Ausgabe

Hier sind zwei Beispiele für XSLT-Stylesheets, die sich zum Testen von XML-Ausgabe eignen.

- Das folgende Stylesheet-Beispiel gibt die angeforderten XML-Daten ohne Umwandlung aus. Dieses Stylesheet ist nützlich für die Anzeige der tatsächlichen XML-Daten, die die Web Publishing Engine verwendet.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <xsl:copy-of select="."/>
  </xsl:template>
</xsl:stylesheet>
```

- Bei der Fehlerbehebung in einem Stylesheet können Sie das folgende Beispiel des HTML-Tags `<textarea>` verwenden, um das XML-Quelldokument anzuzeigen, auf das über das Stylesheet in einem abrollbaren Textbereich zugegriffen wurde. Auf derselben Seite können Sie die umgewandelten XSLT-Ergebnisse mit dem XML-Quelldokument vor der Umwandlung vergleichen.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
  <xsl:output method="html"/>
<html>
  <Datenbereich>
  <xsl:template match="/fmrs:fmresultset">
    <textarea rows="20" cols="100">
      <xsl:copy-of select="."/>
    </textarea><br/>
  </xsl:template>
  </body>
</html>
</xsl:stylesheet>
```

Überwachen Ihrer Website

Sie können anhand der folgenden Typen von Protokolldateien Ihre Custom Web Publishing-Site überwachen und Informationen über Web-Benutzer sammeln, die Ihre Site besuchen:

- Web-Server-Zugriff und Fehlerprotokolle
- Web Publishing Engine-Programmprotokoll
- Web-Server-Modul-Fehlerprotokoll
- interne Web Publishing Core-Zugriffsprotokolle

Verwenden der Web-Server-Zugriffs- und Fehlerprotokolle

Apache (Mac OS): Der Apache-Web-Server generiert eine Zugriffsprotokolldatei und eine Fehlerprotokolldatei. Die Apache-Zugriffsprotokolldatei, standardmäßig im W3C Common Logfile Format, ist eine Aufzeichnung aller beim Web-Server eingehenden HTTP-Abfragen. Das Apache-Fehlerprotokoll ist eine Aufzeichnung von Problemen bei der Verarbeitung von HTTP-Abfragen. Weitere Informationen über diese Protokolldateien finden Sie in der Dokumentation zu Ihrem Apache-Web-Server.

IIS (Windows): Der Microsoft IIS-Web-Server generiert eine Zugriffsprotokolldatei und zeigt Fehler in der Windows-Ereignisanzeige, anstatt sie in eine Protokolldatei zu schreiben. Die Zugriffsprotokolldatei, standardmäßig im W3C Extended Log File Format, ist eine Aufzeichnung aller beim Web-Server eingehenden HTTP-Abfragen. Sie können auch das W3C Common Logfile Format für das Zugriffsprotokoll verwenden. Weitere Informationen finden Sie in der Dokumentation zum Microsoft IIS-Web-Server.

Informationen über das W3C Common Logfile Format und das W3C Extended Log File Format finden Sie auf der Website des World Wide Web Consortium: www.w3.org.

Verwenden des Web Publishing Engine-Programmprotokolls

Standardmäßig generiert die Web Publishing Engine eine Programmprotokolldatei, die eine Aufzeichnung von Web Publishing Engine-Fehler-, Script- und Benutzerprotokollinformationen darstellt.

- Die Fehlerprotokollinformation beschreibt alle ungewöhnlichen Web Publishing Engine-Fehler, die aufgetreten sind. Dem Web-Benutzer häufig gemeldete Fehler wie „Datenbank nicht geöffnet“ werden nicht aufgezeichnet.
- Die Scriptprotokollinformation beschreibt alle Fehler, die generiert wurden, während Web-Benutzer Scripts ausführten. Sie listet z. B. Scriptschritte auf, die übersprungen wurden, weil sie nicht Web-kompatibel sind.
- Die Benutzerprotokollmeldungen umfassen Meldungen, die vom XSLT-Element `<xsl:message>` in XSLT-Stylesheets erzeugt wurden. Immer wenn Web-Benutzer auf Ihr XSLT-Stylesheet zugreifen, werden in der Programmprotokolldatei Informationen aufgezeichnet, die Sie in einem `<xsl:message>`-Element angegeben haben. Weitere Informationen finden Sie unter Kapitel 6, „Entwickeln von FileMaker XSLT-Stylesheets“.

Das Programmprotokoll heißt `pe_application_log.txt` und befindet sich im Ordner Logs, der sich im FileMaker Server-Ordner auf dem Web Publishing Engine-Host befindet.

Die Datei `pe_application_log.txt` wird generiert, wenn beliebige der folgenden Protokollierungsoptionen in der Web Publishing Engine aktiviert sind:

Protokollierungsoption aktiviert	In <code>pe_application_log.txt</code> aufgezeichnete Informationen
Fehlerprotokollierung	Ungewöhnliche Web Publishing Engine-Fehler, die aufgetreten sind. Dem Web-Benutzer häufig gemeldete Fehler wie „Datenbank nicht geöffnet“ werden nicht aufgezeichnet.
Scriptprotokollierung	Fehler, die erzeugt werden, wenn Web-Benutzer Scripts ausführten. Sie listet z. B. Scriptschritte auf, die übersprungen wurden, weil sie nicht Web-kompatibel sind.
Benutzerprotokollierung	Meldungen, die erzeugt werden, wenn Web-Benutzer auf Ihre Custom Web Publishing-Lösung zugreifen.

Alle drei Protokollierungsoptionen sind standardmäßig aktiviert. Informationen zur Einstellung dieser Optionen über die Admin Console finden Sie in der FileMaker Server Hilfe.

Hinweis Die Einträge im Programmprotokoll werden nicht automatisch gelöscht, daher kann die Datei mit der Zeit sehr umfangreich werden. Um Festplattenspeicherplatz auf Ihrem Host-Computer zu sparen, empfiehlt sich ein regelmäßiges Archivieren der Programmprotokolldatei.

Verwenden des Web-Server-Modul-Fehlerprotokolls

Wenn der Web-Server keine Verbindung zur Web Publishing Engine aufbauen kann, generiert das Web Server Modul eine Protokolldatei, die alle Fehler in Zusammenhang mit ihrer Ausführung aufzeichnet. Diese Datei heißt `web_server_module_log.txt` und befindet sich im Ordner Logs im FileMaker Server-Ordner auf dem Web-Server-Host.

Verwenden von internen Web Publishing Core-Zugriffsprotokollen

Die Softwarekomponente Web Publishing Core der Web Publishing Engine generiert zwei interne Zugriffsprotokolldateien, die jeden Zugriff auf den Web Publishing Core aufzeichnen.

- Das Zugriffsprotokoll `wpc_access_log.txt` enthält eine Aufzeichnung aller Endanwenderabfragen zur Generierung von XML und Verwendung von FileMaker Server Instant Web Publishing. Diese Abfragen werden vom Web-Server direkt an den Web Publishing Core geleitet.
- Das Zugriffsprotokoll `pe_internal_access_log.txt` enthält eine Aufzeichnung aller internen XML-Abfragen, die die Softwarekomponente XSLT-CWP der Web Publishing Engine ausführt, während XSLT-Abfragen verarbeitet werden. Diese Abfragen werden intern in der Web Publishing Engine von der Softwarekomponente XSLT-CWP an die Softwarekomponente Web Publishing Core geleitet.

Diese Protokolldateien befinden sich im Ordner Logs, der sich im FileMaker Server-Ordner auf dem Web Publishing Engine-Host befindet.

Die internen Zugriffsprotokolle werden generiert, wenn die Option Anmeldungsprotokollierung in der Web Publishing Engine aktiviert ist. Die Standardeinstellung ist aktiviert. Informationen zur Einstellung der Option Anmeldungsprotokollierung in Admin Console finden Sie in der FileMaker Server Hilfe.

Anhang A

Gültige Namen in Query-Strings

Dieser Anhang beschreibt die gültigen Namen für Query-Befehle und deren Parameter, die Sie in einem Query-String verwenden können, wenn Sie über die Web Publishing Engine auf FileMaker-Daten zugreifen.

Erläuterung von Query-Befehlen und -Parametern

Die folgende Liste enthält sämtliche Namen für Query-Befehle und Query-Parameter:

Query-Befehlsnamen	Query-Parameternamen
–dbnames (siehe Seite 97.)	–db (siehe Seite 101.)
–delete (siehe Seite 97.)	–encoding (nur XSLT) (siehe Seite 101.)
–dup (siehe Seite 98.)	–field (siehe Seite 101.)
–edit (siehe Seite 98.)	fieldname (siehe Seite 102.)
–find, –findall, –findany (siehe Seite 98.)	fieldname.op (siehe Seite 103.)
–findquery (siehe Seite 99.)	–grammar (nur XSLT) (siehe Seite 104.)
–layoutnames (siehe Seite 99.)	–lay (siehe Seite 104.)
–new (siehe Seite 99.)	–lay.response (siehe Seite 104.)
–process (nur XSLT) (siehe Seite 100.)	–lop (siehe Seite 104.)
–scriptnames (siehe Seite 100.)	–max (siehe Seite 105.)
–view (siehe Seite 100.)	–modid (siehe Seite 105.)
	–query (siehe Seite 105.)
	–recid (siehe Seite 106.)
	–relatedsets.filter (siehe Seite 106.)
	–relatedsets.max (siehe Seite 107.)
	–script (siehe Seite 107.)
	–script.param (siehe Seite 108.)
	–script.prefind (siehe Seite 108.)
	–script.prefind.param (siehe Seite 108.)
	–script.presort (siehe Seite 109.)
	–script.presort.param (siehe Seite 109.)
	–skip (siehe Seite 109.)
	–sortfield.[1-9] (siehe Seite 110.)
	–sortorder.[1-9] (siehe Seite 110.)
	–stylehref (siehe Seite 111.)
	–styletype (siehe Seite 111.)
	–token.[string] (nur XSLT) (siehe Seite 111.)

Wichtig Der Parameter –lay für die Angabe eines Datenbanklayouts ist bei allen Query-Befehlen erforderlich mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process (nur XSLT-Abfragen).

Richtlinien für die Verwendung von Query-Befehlen und -Parametern

Beachten Sie folgende Richtlinien, wenn Sie Query-Befehle und -Parameter in einem Query-String verwenden:

- Ein Query-String darf nur einen einzigen Query-Befehl enthalten. Ein Query-String kann z. B. `-new` enthalten, um einen neuen Datensatz hinzuzufügen, aber derselbe Query-String kann nicht gleichzeitig `-new` und `-edit` enthalten.
- Die meisten Query-Befehle verlangen mehrere entsprechende Query-Parameter im Query-String. Beispielsweise verlangen alle Query-Befehle außer `-dbnames` und `-process` den Parameter `-db`, der die abzufragende Datenbank angibt. (Siehe dazu die Tabellen der erforderlichen Parameter in „Abfragen von XML-Daten mithilfe von FileMaker-Query-Strings“ auf Seite 50).
- Geben Sie für Query-Parameter und Feldnamen den jeweils gewünschten Wert an, z. B. `-db=employees`. Geben Sie für Query-Befehle kein `=`-Zeichen oder einen Wert nach dem Befehlsnamen wie `-findall` an.
- Query-Befehls- und -Parametername müssen in Kleinbuchstaben, z. B. `-delete` oder `-lay`, angegeben werden.
- Datenbanknamen, Layoutnamen und Feldnamen, die in Query-Strings verwendet werden, unterscheiden keine Groß- und Kleinschreibung, z. B. gibt `-lay=meinlayout` den Layoutnamen `MeinLayout` an.

Hinweis Feld- und Datenbanknamen, die in XSLT-Anweisungen außerhalb von Query-Strings verwendet werden, unterscheiden Groß-/Kleinschreibung und müssen exakt mit den tatsächlich in der Datenbank verwendeten Namen übereinstimmen. In dieser Anweisung:

```
<xsl:value-of select="fmrs:field[@name='LastName']"/>
```

Der Feldverweis `LastName` in diesem Beispiel muss exakt mit dem Namen des Felds `LastName` in der Datenbank übereinstimmen.

- Feldnamen können Punkte enthalten, es gelten jedoch folgende Ausnahmen:
 - Dem Punkt darf keine Zahl folgen. Beispielsweise ist `meinfeld.9` ein ungültiger Dateiname.
 - Dem Punkt darf nicht die Textzeichenfolge `op` (die beiden Buchstaben `o` und `p`) folgen. Beispielsweise ist `meinfeld.op` ein ungültiger Feldname.
 - Dem Punkt darf nicht die Textzeichenfolge `global` (das Wort „global“) folgen. Beispielsweise ist `meinfeld.global` ein ungültiger Feldname.

Auf Feldnamen, die eine dieser Ausnahmen enthalten, kann nicht über XML oder XSLT mithilfe einer HTTP-Query zugegriffen werden. Diese Konstrukte sind für Datensatz-IDs wie in Abschnitt „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“, nachfolgend beschrieben reserviert.

- Für den Befehl `-find` wird die Groß-/Kleinschreibung eines Feldwerts nicht berücksichtigt. Sie können beispielsweise `Field1=Blue` oder `Field1=blue` verwenden. Für die Befehle `-new` und `-edit` wird die verwendete Schreibung für einen Feldwert beibehalten und exakt so in der Datenbank gespeichert wie im Query-String angegeben. Beispiel: `LastName=Doe`.

Erläuterung der Referenz von FileMaker-Query-Strings

Diese Version enthält eine FileMaker-Datenbank mit dem Namen `Query Strings Reference.fp7`, die kurze Beschreibungen und Beispiele für jeden der FileMaker-Query-Befehle und -Query-Parameter bietet. Diese finden Sie im folgenden Verzeichnis auf jedem Rechner Ihres FileMaker Server-Einsatzes (Master- oder Arbeitscomputer):

Mac:

```
/Library/FileMaker Server/Examples/XSLT
```

Windows:

```
<Laufwerk>\Programme\FileMaker\FileMaker Server\Examples\XSLT
```

Wo: <Laufwerk> das Hauptlaufwerk ist, von dem das System startet.

Erläuterung der Syntax für einen voll qualifizierten Feldnamen

Ein vollständig qualifizierter Feldname identifiziert eine exakte Instanz eines Felds. Da Felder mit gleichen Namen auf unterschiedlichen Tabellen basieren können, müssen Sie vollständig qualifizierte Namen verwenden, um Fehler zu vermeiden.

Syntax für einen vollständig qualifizierten Feldnamen:

tabellen-name::feld-name(wiederholung-nummer).datensatz-id

Dabei gilt:

- **tabellen-name** ist der Name der Tabelle, die das Feld enthält. Der Tabellename ist nur erforderlich, wenn sich das Feld nicht in der zugrunde liegenden Tabelle des im Query-String angegebenen Layouts befindet.
- **feld-name(wiederholung-nummer)** ist der spezifische Wert in einem Wiederholfeld und ist nur für Wiederholfelder erforderlich. Die Wiederholungsnummer wird ab der Ziffer 1 gezählt. Beispielsweise gibt **feld-name(2)** den zweiten Wert in einem Wiederholfeld an. Wenn Sie keine Wiederholungsnummer für ein Wiederholfeld angeben, wird der erste Wert des Wiederholfelds verwendet. Die Wiederholungsnummer ist für die Query-Befehle `-new` und `-edit` erforderlich, wenn sie Wiederholfelder verwenden, jedoch nicht für den Befehl `-find`.
- **datensatz-id** gibt die Datensatz-ID an. Diese ist nur erforderlich, wenn Sie einen Query-String verwenden, um Datensätze in einem Ausschnitt hinzuzufügen oder zu bearbeiten. Weitere Informationen finden Sie in den Abschnitten „Hinzufügen von Datensätzen in einem Ausschnitt“ und „Bearbeiten von Datensätzen in einem Ausschnitt“. Die **record-id** ist für die Query-Befehle `-new` und `-edit` erforderlich, wenn sie Ausschnittfelder verwenden, jedoch nicht für den Befehl `-find`.

Hinweis Damit sie verfügbar sind, müssen Felder auf das Layout gesetzt werden, das Sie im Query-String angeben.

Verwenden von Query-Befehlen für Ausschnittfelder

Die folgenden Abschnitte beschreiben, wie Query-Befehle für Ausschnittfelder funktionieren.

Hinzufügen von Datensätzen in einem Ausschnitt

Um einem Ausschnitt gleichzeitig mit dem Hinzufügen eines Ausgangsdatsatzes einen neuen Datensatz hinzuzufügen, verwenden Sie den Query-Befehl `-new` und führen den folgenden Query-String für die Abfrage aus:

- Verwenden Sie den vollständig qualifizierten Feldnamen für das Bezugsfeld im Ausschnitt.
- Geben Sie 0 als Datensatz-ID nach dem Namen des Bezugsfelds im Ausschnitt an.
- Geben Sie mindestens eines der Felder für den Ausgangsdatsatz an, bevor Sie das Bezugsfeld im Ausschnitt angeben.
- Geben Sie die Daten für das Abgleichsfeld (Schlüsselfeld) im Ausgangsdatsatz an.

Beispiel: Der folgende URL fügt einen neuen Employee-Ausgangsdatsatz für John Doe und gleichzeitig einen neuen Bezugsdatensatz für Jane im Ausschnitt hinzu. Der Name der Bezugstabelle ist „Dependents“ und der Name des Bezugsfelds im Ausschnitt lautet „Names“. Das Abgleichsfeld ID speichert eine Angestelltenkennung.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family&FirstName=John&LastName=Doe
&ID=9756&Dependents::Names.0=Jane&-new
```

Hinweis Sie können pro Abfrage nur einen Bezugsdatensatz hinzufügen.

Bearbeiten von Datensätzen in einem Ausschnitt

Zur Bearbeitung eines oder mehrerer Datensätze in einem Ausschnitt verwenden Sie den Befehl `-edit` und eine Datensatz-ID, um den Ausgangsdatensatz anzugeben, der die zu bearbeitenden Ausschnitt-Datensätze enthält. Geben Sie den entsprechenden zu bearbeitenden Ausschnitt-Datensatz anhand seiner Datensatz-ID in einem vollständig qualifizierten Feldnamen an. Sie können eine Datensatz-ID dem Datensatz-ID-Attribut des `<record>`-Elements im `<relatedset>`-Element in den XML-Daten entnehmen. Weitere Informationen finden Sie unter „Verwenden der `fmresultset`-Grammatik“ auf Seite 41

Beispiel: Der folgende URL bearbeitet einen Datensatz in einem Ausschnitt, wobei der Ausgangsdatensatz die Datensatz-ID 1001 hat. Der Name der Bezugstabelle ist „Dependents“, der Name des Bezugsfelds im Ausschnitt lautet „Names“ und die 2 in „Names.2“ ist die Datensatz-ID des Ausschnitt-Datensatzes.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family&-recid=1001
&Dependents::Names.2=Kevin&-edit
```

Das folgende Beispiel zeigt, wie Sie eine Abfrage verwenden können, um mehrere Ausschnitt-Datensätze über den Ausgangsdatensatz zu bearbeiten:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family&-recid=1001
&Dependents::Names.2=Kevin&Dependents::Names.5=Susan&-edit
```

Sie können auch den Befehl `-edit` verwenden und 0 als ID des Ausschnitt-Datensatzes angeben, um dem Ausschnitt einen neuen Bezugsdatensatz für einen bestehenden Ausgangsdatensatz hinzuzufügen. Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family&-recid=1001
&Dependents::Names.0=Timothy&-edit
```

Löschen von Ausschnittdatensätzen

Um Ausschnittdatensätze zu löschen, verwenden Sie den Parameter `-delete.related` mit dem Befehl `-edit` anstelle des Befehls `-delete`.

Zum Beispiel löscht der folgende URL den Datensatz 1001 aus der Tabelle `employees`:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family&-recid=1001&-delete
```

Aber der folgende URL löscht einen Ausschnittdatensatz mit einer Datensatz-ID 3 aus der Bezugstabelle `Dependents` mit der Ausgangsdatensatz-ID 1001.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family&-recid=1001
&-delete.related=Dependents.3&-edit
```

Weitere Informationen finden Sie unter „`-delete.related` (Löschen von Ausschnittdatensätzen, Query-Parameter)“ auf Seite 101

Abfragen von Ausschnittfeldern

In einer Lösung mit vielen Datensätzen kann das Abfragen und Sortieren von Datensätzen einige Zeit in Anspruch nehmen. Um die anzuzeigende Anzahl von Datensätzen und Zeilen in einer Bezugsmenge einzuschränken, verwenden Sie die Parameter `-relatedsets.filter` und `-relatedsets.max` bei Suchabfragen. Weitere Informationen finden Sie unter „`-relatedsets.filter` (Ausschnittdatensätze filtern, Query-Parameter)“ auf Seite 106 und „`-relatedsets.max` (Ausschnittdatensätze beschränken, Query-Parameter)“ auf Seite 107

Erläuterung der Syntax für das Angeben eines Variablenfelds

Syntax für das Angeben eines Variablenfelds:

tabellen-name::feld-name(wiederholung-nummer).global

Dabei gilt: `global` gibt ein Feld an, das global gespeichert wird. Informationen über `tabellen-name` und `feld-name(wiederholung-nummer)` finden Sie unter „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 95. Informationen über Variablenfelder finden Sie in der FileMaker Pro Hilfe.

Sie müssen die `.global`-Syntax verwenden, um ein Variablenfeld in einem Query-String zu identifizieren. Die Web Publishing Engine setzt die Parameterwerte für Variablenfelder, bevor sie den Query-Befehl ausführt oder andere Parameterwerte im Query-String setzt. Für direkte XML-Abfragen sowie Abfragen, die über XSLT-Stylesheets erfolgen und keine Sitzungen verwenden, laufen die globalen Werte sofort nach Ausführung der Abfrage ab. Für Abfragen, die über XSLT-Stylesheets erfolgen und Sitzungen verwenden, bleiben die globalen Werte für die Dauer der Sitzung erhalten, die im Stylesheet definiert ist, oder bis sie erneut durch eine andere Abfrage geändert werden.

Wenn Sie ein Variablenfeld in einem Query-String nicht mit der `.global`-Syntax identifizieren, berechnet die Web Publishing Engine das Variablenfeld zusammen mit dem restlichen Query-String, ohne zuvor den Wert für das Variablenfeld zu setzen.

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments
&Country.global=USA&-recid=1&-edit
```

Wichtig Wenn Sie ein Variablenfeld in einem XSLT-Stylesheet benutzen, müssen Sie mithilfe von Admin Console die Option XSLT-Datenbanksitzungen für die Web Publishing Engine aktivieren. Andernfalls werden Variablenfeldwerte zwischen Abfragen nicht beibehalten. Weitere Informationen finden Sie in der FileMaker Server Hilfe.

Query-Befehlsreferenz

Dieser Abschnitt enthält Informationen über die Query-Befehle, die für XML- und XSLT-Abfragen zur Verfügung stehen.

Hinweis Nur in XSLT-Abfragen erfordern alle folgenden Query-Befehle den Query-Parameter `-grammar`.

Query-Befehl `-dbnames` (Datenbanknamen)

Ruft die Namen aller Datenbanken ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML oder XSLT aktiviert sind.

Erforderliche Query-Parameter: (keine)

Beispiel:

So rufen Sie die Datenbanknamen ab:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-dbnames
```

Query-Befehl `-delete` (Datensatz löschen)

Löscht den vom `-recid`-Parameter angegebenen Datensatz.

Erforderliche Query-Parameter: `-db`, `-lay`, `-recid`

Optionaler Query-Parameter: `-script`

Beispiel:

So löschen Sie einen Datensatz:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-recid=4&-delete
```

Query-Befehl –dup (Datensatz duplizieren)

Dupliziert den durch –recid angegebenen Datensatz.

Erforderliche Query-Parameter: –db, –lay, –recid

Optionaler Query-Parameter: –script

Beispiel:

So duplizieren Sie den angegebenen Datensatz:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–recid=14&–dup`

Query-Befehl –edit (Datensatz bearbeiten)

Aktualisiert den vom Parameter –recid angegebenen Datensatz und füllt die Felder mit dem Inhalt etwaiger Feldname-Wert-Paare. Der –recid-Parameter gibt an, welcher Datensatz bearbeitet werden sollte.

Erforderliche Query-Parameter: –db, –lay, –recid, ein oder mehrere Feldname(n)

Optionaler Query-Parameter: –modid, –script

Hinweis Informationen über das Bearbeiten von Datensätzen in einem Ausschnitt finden Sie unter „Bearbeiten von Datensätzen in einem Ausschnitt“ auf Seite 96

Beispiel:

So bearbeiten Sie einen Datensatz:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–recid=13&Country=USA&–edit`

Query-Befehle –find, –findall oder –findany (Datensätze suchen)

Sendet eine Suchabfrage mit definierten Kriterien.

Erforderliche Query-Parameter: –db, –lay

Optionale Query-Parameter: –recid, –lop, –op, –max, –skip, –sortorder, –sortfield, –script, –script.prefind, –script.presort, Feldname

Beispiele:

So suchen Sie einen Datensatz nach Feldname:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&Country=USA&–find`

Hinweis Die mehrfache Angabe eines Feldnamens in einer einzelnen Abfrage wird nicht unterstützt. FileMaker Server geht alle Werte durch, verwendet aber nur den zuletzt gearperten Wert.

So suchen Sie einen Datensatz nach Datensatz-ID:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–recid=427&–find`

Um alle Datensätze in der Datenbank zu finden, verwenden Sie –findall:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–findall`

Um einen zufällig ermittelten Datensatz zu finden, verwenden Sie –findany:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=family&–findany`

-findquery (Compound-Suche, Query-Befehl)

Sendet eine Suchabfrage mit mehreren Suchdatensätzen und Datensatzausschluss-Abfragen.

Erforderliche Query-Parameter: -db, -lay, -query

Optionale Query-Parameter: -max, -skip, -sortorder, -sortfield, -script, -script.prefind, -script.presort

Beispiel:

Sucht Datensätze mit Hunden oder Katzen, die nicht „Fluffy“ heißen.

```
http://host/fmi/xml/fmresultset.xml?-db=vetclinic&-lay=animals&-query=(q1);(q2);!(q3)
&-q1=typeofanimal&-q1.value=Cat&-q2=typeofanimal&-q2.value=Dog&-q3=name&-q3.value=Fluffy&-findquery
```

Verwenden des Befehls -findquery für Compound-Suchen

Eine Anweisung -findquery besteht aus vier Teilen in der folgenden Reihenfolge:

- Der Parameter -query
- Die Query-Abfragedeclarationen, bestehend aus den Query-Bezeichnerdeklarationen und Abfragevorgängen
- Das Suchfeld und Wertedefinitionen für jeden Query-Bezeichner
- Der Befehl -findquery am Ende der gesamten Anweisung

Weitere Informationen zur Verwendung des Parameters -query finden Sie unter „-query (Compound-Suchabfrage, Query-Parameter)“ auf Seite 105

-layoutnames (Layoutnamen) (Query-Befehl)

Ruft die Namen aller verfügbaren Layouts für eine angegebene Datenbank ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML oder XSLT aktiviert sind.

Erforderliche Query-Parameter: -db

Beispiel:

So rufen Sie die Namen verfügbarer Layouts ab:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-layoutnames
```

-new (Neuer Datensatz) (Query-Befehl)

Erzeugt einen neuen Datensatz und fügt in diesen den Inhalt jeglicher Feldnamen/Wert-Paare ein.

Erforderliche Query-Parameter: -db, -lay

Optionalen Query-Parameter: ein oder mehrere Feldname(n), -script

Hinweis Informationen über das Einfügen neuer Daten für einen Ausschnitt finden Sie unter „Hinzufügen von Datensätzen in einem Ausschnitt“ auf Seite 95

Beispiel:

So fügen Sie einen neuen Datensatz hinzu:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&Country=Australia&-new
```

-process (XSLT-Stylesheets verarbeiten)

Verarbeitet ein XSLT-Stylesheet, ohne Daten von der Datenbank anzufordern. Dieser Query-Befehl kann nur mit XSLT-Stylesheets verwendet werden.

Erforderlicher Query-Parameter: –grammar Sie müssen die fmresultset- oder FMPXMLRESULT-Grammatik verwenden.

Beispiel:

`http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xml?–grammar=fmresultset&–process`

Weitere Informationen finden Sie unter „Verarbeitung von XSLT-Abfragen, die nicht FileMaker Server abfragen“ auf Seite 62

–scriptnames (Scriptnamen) (Query-Befehl)

Ruft die Namen aller verfügbaren Scripts für eine angegebene Datenbank ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML oder XSLT aktiviert sind.

Erforderliche Query-Parameter: –db

Beispiel:

So rufen Sie die Namen aller Scripts ab:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–scriptnames`

Query-Befehl –view (Layoutinformationen anzeigen)

Wenn die FMPXMLLAYOUT-Grammatik angegeben ist: Ruft Layoutinformationen aus einer Datenbank ab und zeigt sie in der FMPXMLLAYOUT-Grammatik an. Wenn eine Datengrammatik (fmresultset oder FMPXMLRESULT) angegeben ist: Ruft den Metadaten-Abschnitt des XML-Dokuments und einen leeren „recordset“ ab.

Erforderliche Query-Parameter: –db, –lay

Optionaler Query-Parameter: –script

Beispiele:

So rufen Sie Layoutinformationen ab:

`http://192.168.123.101/fmi/xml/FMPXMLLAYOUT.xml?–db=employees&–lay=departments&–view`

So rufen Sie Metadaten-Informationen ab:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–view`

Query-Parameterreferenz

Dieser Abschnitt enthält Informationen über die Query-Parameter, die für XML- und XSLT-Abfragen zur Verfügung stehen. Informationen über Parameter, die nur für XSLT-Abfragen verfügbar sind, finden Sie unter „Verwenden von Query-Strings in FileMaker XSLT-Stylesheets“ auf Seite 58

Query-Parameter `-db` (Datenbankname)

Gibt die Datenbank an, auf die der Query-Befehl angewendet wird.

Wert: Name der Datenbank ohne etwaige Dateinamenerweiterung.

Hinweis Geben Sie mit dem Namen der Datenbank für den Parameter `-db` keine Dateinamenerweiterung an. Der tatsächliche Datenbank-Dateiname kann optional eine Erweiterung aufweisen, aber Erweiterungen sind als Wert für den Parameter `-db` nicht erlaubt.

Erforderlich für: Alle Query-Befehle mit Ausnahme von `-dbnames` und `-process`

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-findall
```

`-delete.related` (Löschen von Ausschnitt Datensätzen, Query-Parameter)

Löscht einen Datensatz aus einem Ausschnittfeld.

Optional für: `-edit` (Query-Befehl)

Erfordert: Ein Bezugstabellenname und eine Datensatz-ID

Beispiel:

Das folgende Beispiel löscht einen Ausschnitt Datensatz mit einer Datensatz-ID 20 aus der Bezugstabelle `jobtable` mit der Ausgangsdatsatz-ID 7.

```
http://host/fmi/xml/fmresultset.xml?-db=career&-lay=applications&-recid=7&-delete.related=jobtable.20&-edit
```

Query-Parameter `-encoding` (XSLT-Abfrage-Kodierung)

Gibt die Textkodierung für eine XSLT-Abfrage an. Dieser Query-Befehl kann nur für Custom Web Publishing mit XSLT-Abfragen verwendet werden.

Wert: US-ASCII, ISO-8859-1, ISO-8859-15, ISO-2022-JP, Shift_JIS, or UTF-8

Optional für: Alle Query-Befehle in einer XSLT-Abfrage

Beispiel:

```
http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xml?-db=employees&-lay=departments
&-grammar=fmresultset&-encoding=Shift_JIS&-findall
```

Weitere Informationen finden Sie unter „Festlegen von Textkodierung für Abfragen“ auf Seite 61

Query-Parameter `-field` (Medienfeldname)

Gibt den Namen eines Medienfelds zurück.

Erforderlich für: Abfrage von Daten in einem Medienfeld.

Weitere Informationen finden Sie unter „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XML-Lösungen“ auf Seite 38, und „Erläuterung der URL-Syntax für FileMaker-Medienobjekte in XSLT-Lösungen“ auf Seite 57

Query-Parameter „fieldname“ (Nicht-Medienfeldname)

Feldnamen dienen als Kontrollkriterien für den Query-Befehl `-find` oder zum Ändern des Inhalts eines Datensatzes. Wenn ein Query-Befehl oder -Parameter einen Wert für ein Nicht-Medienfeld verlangt, verwenden Sie den Feldnamen ohne den Bindestrich (`-`) als den Namensteil des Name-Wert-Paars.

Name: Name des Felds in der FileMaker-Datenbank. Wenn das Feld sich nicht in der zugrunde liegenden Tabelle des Layouts befindet, das im Query-String angegeben ist, muss der Feldname vollständig qualifiziert sein. Feldnamen können Punkte enthalten, es gelten jedoch folgende Ausnahmen:

- Dem Punkt darf keine Zahl folgen. Beispielsweise ist `meinfeld.9` ein ungültiger Dateiname.
- Dem Punkt darf nicht die Textzeichenfolge `op` (die beiden Buchstaben `o` und `p`) folgen. Beispielsweise ist `meinfeld.op` ein ungültiger Feldname.
- Dem Punkt darf nicht die Textzeichenfolge `global` (das Wort „global“) folgen. Beispielsweise ist `meinfeld.global` ein ungültiger Feldname.

Auf Feldnamen, die eine dieser Ausnahmen enthalten, kann nicht über XML oder XSLT mithilfe einer HTTP-Query zugegriffen werden. Diese Konstrukte sind für Datensatz-IDs wie in Abschnitt „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 95 beschrieben reserviert.

Wert: Geben Sie für die Query-Befehle `-new` und `-edit` den Wert an, den Sie im aktuellen Datensatz in dem Feld speichern wollen. Geben Sie für den Query-Befehl `-find` den Wert an, den Sie im Feld suchen wollen. Wenn Sie den Wert für ein Datums-, Zeit- oder Zeitstempelfeld angeben, müssen Sie den Wert gemäß „fm“-Format für den Feldtyp angeben. Die „fm“-Formate sind `MM/tt/jjjj` für Datum, `HH:mm:ss` für Zeit und `MM/tt/jjjj HH:mm:ss` für Zeitstempel.

Erforderlich für: `-edit` (Query-Befehl)

Optional für: `-new`- und `-find`-Query-Befehle

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-op=eq&FirstName=Sam
&-max=1&-find
```

Hinweis Die mehrfache Angabe eines Feldnamens in einer einzelnen Abfrage wird nicht unterstützt. FileMaker Server geht alle Werte durch, verwendet aber nur den zuletzt geparsten Wert.

Query-Parameter „fieldname.Op“ (Vergleichsoperator)

Gibt den Vergleichsoperator an, der auf den Feldnamen, der dem Operator vorangeht, angewendet werden soll. Vergleichsoperatoren werden mit dem Query-Befehl `-find` verwendet.

Wert: Der zu verwendende Operator. Der Standard-Operator lautet „begins with“. Gültige Operatoren sind:

Schlüsselwort	Äquivalenter Operator in FileMaker Pro
eq	= Wort
cn	*Wort*
bw	Wort*
ew	*Wort
gt	> Wort
gte	>= Wort
lt	< Wort
lte	<= Wort
neq	ausschließen, Wort

Optional für: `-find` (Query-Befehl)

Erfordert: Einen Feldnamen und einen Wert

Syntax für das Angeben eines Vergleichsoperators:

tabellen-name::feld-name=wert&tabelle-name::feld-name.op=op-symbol

Dabei gilt:

- `tabelle-name` ist die Tabelle, die das Feld enthält, und ist nur erforderlich, wenn das Feld nicht in der Quelltable des im Query-String angegebenen Layouts ist.
- `op-symbol` ist eines der Schlüsselwörter in der vorherigen Tabelle, wie z. B. `cn`.

Beispiel:

`http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&name=Tim&name.op=cn&-find`

Sie können jeden FileMaker Pro-Suchoperator verwenden, indem Sie das Schlüsselwort `bw` angeben.

Um z. B. den Wertebereich über den Bereichsoperator (...) zu ändern, geben Sie das Schlüsselwort `bw` an und setzen dann die Zeichen „...“ vor das Suchkriterium.

Beispiel:

`http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&IDnum=915...925&IDnum.op=bw&-find`

Weitere Informationen über Operatoren für die Suche nach Text finden Sie in der FileMaker Pro Hilfe.

Query-Parameter –grammar (Grammatik für XSLT-Stylesheets)

Gibt die Grammatik für ein XSLT-Stylesheet an. Dieser Query-Befehl kann nur für Custom Web Publishing mit XSLT-Abfragen verwendet werden.

Wert: fmresultset, FMPXMLRESULT oder FMPXMLLAYOUT

Erforderlich für: Alle XSLT-Abfragen

Beispiel:

```
http://192.168.123.101/fmi/xsl/my_template/my_stylesheet.xml?–grammar=fmresultset&–db=mydatabase  
&–lay=mylayout&–findall
```

Weitere Informationen finden Sie unter „Angabe einer XML-Grammatik für ein FileMaker XSLT-Stylesheet“ auf Seite 58

Query-Parameter –lay (Layout)

Gibt das Datenbanklayout an, das Sie verwenden wollen.

Wert: Name des Layouts.

Erforderlich für: Alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process (nur XSLT-Abfragen)

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–view
```

Query-Parameter –lay.response (Layout wechseln für Antwort)

Gibt an, dass FileMaker Server beim Verarbeiten einer Abfrage das Layout verwenden soll, das der Parameter –lay angibt, und bei der Verarbeitung der XML-Antwort in das Layout wechseln soll, das der Parameter –lay.response angibt.

Wenn Sie den Parameter –lay.response nicht angeben, verwendet FileMaker Server das vom Parameter –lay angegebene Layout für die Verarbeitung der Abfrage und der Antwort.

Sie können den Parameter –lay.response für XML-Abfragen oder in XSLT-Stylesheet-Abfragen verwenden.

Wert: Name des Layouts.

Optional für: Alle Query-Befehle mit Ausnahme von –dbnames, –layoutnames, –scriptnames und –process (nur XSLT-Abfragen)

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=Budget&Salary=100000&Salary.op=gt&–find  
&–lay.response=VorstListe
```

Query-Parameter –lop (Logischer Operator)

Gibt an, wie die Suchkriterien im Query-Befehl –find als and- oder or-Suche kombiniert werden.

Wert: and oder or (müssen in Kleinbuchstaben angegeben werden). Wenn der Query-Parameter –lop nicht angegeben wird, verwendet der Query-Befehl –find den Wert and.

Optional für: –find (Query-Befehl)

Hinweis Nicht unterstützt vom Query-Befehl -findquery.

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&Last+Name=Smith  
&Birthdate=2/5/1972&–lop=and&–find
```

Query-Parameter –max (Maximum Datensätze)

Gibt die maximale Anzahl an Datensätzen an, die zurückgegeben werden soll.

Wert: Eine Zahl oder – für die Rückgabe aller Datensätze – der Wert all. Der Wert all muss in Kleinbuchstaben angegeben werden. Wenn –max nicht angegeben wird, werden alle Datensätze zurückgegeben.

Optional für: Query-Befehle –find, –findall und –findquery

Beispiele:

`http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-max=10&-findall`

`http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-max=all&-findall`

Query-Parameter –modid (Änderungs-ID)

Die Änderungs-ID ist ein inkrementeller Zähler, der die aktuelle Version eines Datensatzes angibt. Indem Sie mit dem Query-Befehl –edit eine Änderungs-ID angeben, können Sie sicherstellen, dass Sie die aktuelle Version eines Datensatzes bearbeiten. Wenn der angegebene Wert der Änderungs-ID nicht dem aktuellen Wert der Änderungs-ID in der Datenbank entspricht, wird der Query-Befehl –edit nicht zugelassen und ein Fehlercode wird zurückgegeben.

Wert: Eine Änderungs-ID, die die aktuelle Version eines Datensatzes in einer FileMaker-Datenbank eindeutig identifiziert.

Optional bei: –edit (Query-Befehl)

Erfordert: –recid-Parameter

Beispiel:

`http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-recid=22&-modid=6
&last_name=Jones&-edit`

–query (Compound-Suchabfrage, Query-Parameter)

Gibt die Query-Namen und Suchkriterien für eine Compound-Suchabfrage an. Weitere Informationen finden Sie unter „–findquery (Compound-Suche, Query-Befehl)“ auf Seite 99

Wert: Ein Query-Ausdruck.

Erforderlich für: Query-Befehl –findquery

Die Syntax für eine Compound-Suchabfrage ist:

`–query=<request-declarations><request-definitions>&–findquery`

Wo:

`<request-declarations>` sind zwei oder mehr Abfragedeklarationen.

- Jede Abfragedeklaration setzt sich aus einem oder mehreren Query-Bezeichnern zusammen, die durch Kommata getrennt und in Klammern eingeschlossen sind. Ein Query-Bezeichner ist der Buchstabe „q“ gefolgt von einer Zahl. Beispiel: q1
- Eingeschlossen in Klammern fungieren mehrere Queries als logische UND-Suchen, die die Ergebnismenge einengen. Zum Beispiel gibt (q1, q2) Datensätze zurück, die q1 und q2 entsprechen.

- Wie bei FileMaker Pro kann jede Abfrage eine Such- oder Ausschlussabfrage sein. Eine Suchabfrage fügt die entsprechenden Datensätze der Ergebnismenge hinzu. Eine Ausschlussabfrage entfernt die entsprechenden Datensätze aus der Ergebnismenge. Standard ist eine Suchabfrage. Setzen Sie für eine Ausschlussabfrage ein Ausrufezeichen (!) vor die öffnende Klammer.

Beispiel: (q1)!(q2)

In diesem Beispiel ist q1 eine Suchabfrage. q2 ist eine Ausschlussabfrage, da ihr ein Ausrufezeichen vorangestellt ist.

- Abfragen werden durch Semikolon getrennt. Mehrere Suchabfragen fungieren als logische ODER-Suchen, die die Ergebnismenge erweitern. Zum Beispiel gibt (q1);(q2) Datensätze zurück, die q1 oder q2 entsprechen. Ausschlussabfragen fungieren nicht als logische ODER-Suchen, da Ausschlussabfragen Datensätze aus der Ergebnismenge entfernen.
- Abfragen werden in der angegebenen Reihenfolge ausgeführt. Die Ergebnismenge umfasst die Ergebnisse aus der gesamten zusammengesetzten Suchabfrage.

<request-definitions> ist eine Abfragedefinition für jede Abfragedeklaration. Jede Abfragedefinition besteht aus Suchfeld und Wertdefinition. Ein Minuszeichen (–) beginnt die Abfragedefinition.

Syntax:

–<query-id>=<Feldname>&–<query-id>.value=<Wert>

Beispiel:

–q1=typeofanimal&–q1.value=Cat

–q2=name&–q2.value=Fluffy

Beispiel:

Sucht Datensätze mit grauen Katzen, die nicht „Fluffy“ heißen.

[http://host/fmi/xml/fmresultset.xml?–db=tierklinik&–lay=Patienten&–query=\(q1, q2\)!\(q3\)&–q1=typeofanimal&–q1.value=Cat&–q2=color&–q2.value=Gray&–q3=name&–q3.value=Fluffy&–findquery](http://host/fmi/xml/fmresultset.xml?–db=tierklinik&–lay=Patienten&–query=(q1, q2)!(q3)&–q1=typeofanimal&–q1.value=Cat&–q2=color&–q2.value=Gray&–q3=name&–q3.value=Fluffy&–findquery)

Query-Parameter –recid (Datensatz-ID)

Gibt den Datensatz an, der verarbeitet werden soll. Wird hauptsächlich mit den Query-Befehlen –edit und –delete verwendet. Vom Befehl –view verwendet, um Bezugswertlistendaten in der Grammatik FMPXMLLAYOUT abzurufen.

Wert: Eine Datensatz-ID, die einen Datensatz in einer FileMaker-Datenbank eindeutig angibt.

Erforderlich für: Query-Befehle –edit, –delete und –dup

Optional bei: Befehlen –find query und –view

Beispiel 1:

<http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=departments&–recid=22&–delete>

Beispiel 2:

<http://localhost/fmi/xml/FMPXMLLAYOUT.xml?–db=test&–lay=empty&–view&–recid=9>

–relatedsets.filter (Ausschnitt Datensätze filtern, Query-Parameter)

Gibt an, ob die zurückgegebenen Zeilen für Queries, die Ausschnittfelder verwenden, gefiltert werden.

Wert: layout oder none

- Der Standardwert ist none, wenn dieser Parameter nicht angegeben ist.

- Wenn die Query layout angibt, werden die im FileMaker Pro-Dialogfeld „Ausschnitteinstellungen“ angegebenen Einstellungen respektiert. Die Datensätze werden basierend auf der im Dialogfeld „Ausschnitteinstellungen“ definierten Sortierung sortiert, wobei die Datensatzmenge gefiltert ist, um mit der angegebenen ersten Zeile zu beginnen.
- Wenn die Einstellung „Vertikaler Rollbalken“ im Dialogfeld „Ausschnitteinstellungen“ aktiviert ist, können Sie die Option `-relatedsets.max` verwenden, um die maximale Anzahl an Zeilen anzugeben, die in der Antwort auf die Query zurückgegeben werden.
- Wenn „Vertikaler Rollbalken“ deaktiviert ist, bestimmt die Einstellung „Anzahl der Zeilen“ im FileMaker Pro-Dialogfeld „Ausschnitteinstellungen“ die Anzahl an anzuzeigenden Zeilen.
- Wenn `-relatedsets.filter` auf `none` eingestellt ist, gibt die Web Publishing Engine alle Zeilen im Ausschnitt zurück und die Ausschnittdatensätze werden nicht vorsortiert.

Optional bei: `-find`, `-edit`, `-new`, `-dup` und `-findquery`.

Beispiele:

```
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English&-relatedsets.filter=none&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English&relatedsets.filter=layout
&-relatedsets.max=all&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English&-relatedsets.filter=layout
&-relatedsets.max=10&-findany
```

-relatedsets.max (Ausschnittdatensätze beschränken, Query-Parameter)

Gibt die maximale Anzahl an Zeilen an, die in den Ergebnissen für diese Query zurückgegeben werden.

Wert: eine Ganzzahl oder `all`.

- Wenn die Query eine Ganzzahl angibt, gibt die Web Publishing Engine die Anzahl an Zeilen nach der ersten Zeile zurück.
- Wenn die Query `all` angibt, gibt die Web Publishing Engine alle Bezugsdatensätze zurück.
- Wenn die Query den Parameter `-relatedsets.max` nicht angibt, wird die Anzahl der Zeilen durch den Wert bestimmt, der für den Parameter `-relatedsets.filter` angegeben ist. Weitere Informationen finden Sie unter „`-relatedsets.filter` (Ausschnittdatensätze filtern, Query-Parameter)“ auf Seite 106

Optional für: `-find`, `-edit`, `-new`, `-dup` und `-findquery`.

Beispiele:

```
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English&relatedsets.filter=layout
&-relatedsets.max=all&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English&-relatedsets.filter=layout
&-relatedsets.max=10&-findany
```

-script (Script) (Query-Parameter)

Gibt das FileMaker-Script an, das nach Ausführung des Query-Befehls und der Sortierung ablaufen soll. Weitere Informationen finden Sie unter „Erläuterung der Verarbeitung einer XML-Abfrage“ auf Seite 53

Wert: Scriptname.

Optional für: Alle Query-Befehle mit Ausnahme von `-dbnames`, `-layoutnames`, `-scriptnames` und `-process`

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-script=myscript&-findall
```

Query-Parameter `–script.param` (Parameter an Script übergeben)

Übergibt einen Parameter an das durch `–script` angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. `param1|param2|param3` als Liste mit dem Zeichen `|`, das so kodiert ist: `param1%7Cparam2%7Cparam3`
- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: `LiesAlsZahl (Hole(ScriptParam))`
- Wenn Ihre Query `–script.param` ohne `–script` enthält, wird `–script.param` ignoriert.
- Wenn Ihre Query mehr als einen `–script.param` enthält, verwendet die Web Publishing Engine den letzten geparsten Wert.

Optional für: `–script`

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-script=myscript
&-script.param=Smith%7CChatterjee%7CSu&-findall
```

Query-Parameter `–script.prefind` (Script vor Suchen)

Gibt das FileMaker-Script an, das vor dem Suchen und Sortieren von Datensätzen (falls angegeben) während der Verarbeitung des Query-Befehls `–find` ausgeführt wird.

Wert: Scriptname.

Optional für: Alle Query-Befehle mit Ausnahme von `–dbnames`, `–layoutnames`, `–scriptnames` und `–process`

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-script.prefind=myscript&-findall
```

Query-Parameter `–script.prefind.param` (Parameter an Script vor Suchen übergeben)

Übergibt einen Parameter an das durch `–script.prefind` angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. `param1|param2|param3` als Liste mit dem Zeichen `|`, das so kodiert ist: `param1%7Cparam2%7Cparam3`
- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: `LiesAlsZahl (Hole(ScriptParam))`
- Wenn Ihre Query `–script.prefind.param` ohne `–script.prefind` enthält, wird `–script.prefind.param` ignoriert.
- Wenn Ihre Query mehr als einen `–script.prefind.param` enthält, verwendet die Web Publishing Engine den letzten geparsten Wert.

Optional für: `–script.prefind`

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-script.prefind=myscript
&-script.prefind.param=payroll&-findall
```

Query-Parameter `-script.presort` (Script vor Sortieren)

Gibt das FileMaker-Script an, das nach dem Suchen von Datensätzen (falls angegeben) und vor dem Sortieren der Datensätze während der Verarbeitung des Query-Befehls `-find` ausgeführt wird.

Optional für: Alle Query-Befehle mit Ausnahme von `-dbnames`, `-layoutnames`, `-scriptnames` und `-process`

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-script.presort=myscript
&-sortfield.1=dept&-sortfield.2=rating&-findall
```

Query-Parameter `-script.presort.param` (Parameter an Script vor Sortieren übergeben)

Übergibt einen Parameter an das durch `-script.presort` angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. `param1|param2|param3` als Liste mit dem Zeichen `|`, das so kodiert ist: `param1%7Cparam2%7Cparam3`
- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: `LiesAlsZahl (Hole(ScriptParam))`
- Wenn Ihre Query `-script.presort.param` ohne `-script.presort` enthält, wird `-script.presort.param` ignoriert.
- Wenn Ihre Query mehr als einen `-script.presort.param` enthält, verwendet die Web Publishing Engine den letzten geparsten Wert.

Optional für: `-script.presort`

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-script.presort=myscript
&-script.presort.param=18%7C65&-sortfield.1=dept&-sortfield.2=rating&-findall
```

Query-Parameter `-skip` (Datensätze überspringen)

Gibt an, wie viele Datensätze in der Ergebnismenge übersprungen werden sollen.

Wert: Eine Zahl. Wenn der Wert größer als die Anzahl der Datensätze in der Ergebnismenge ist, wird kein Datensatz angezeigt. Der Standardwert beträgt 0.

Optional für: `-find` (Query-Befehl)

Im folgenden Beispiel werden die ersten 10 Datensätze in der Ergebnismenge ausgelassen und die Datensätze 11 bis 15 ausgegeben.

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-skip=10&-max=5&-findall
```

Query-Parameter –sortfield (Sortierfeld)

Gibt das Feld an, das für das Sortieren verwendet wird.

Wert: Feldname

Optional für: –find- oder –findall-Query-Befehle

Der Query-Parameter –sortfield kann mehrfach verwendet werden, um mehrere Feldsortierungen vorzunehmen. Syntax für die Rangfolge der Sortierfelder:

–sortfield.precedence-number=fully-qualified-field-name

Dabei gilt: rang-nummer im Query-Parameter –sortfield.rang-nummer ist eine Zahl (beginnend mit 1), die die Rangfolge für mehrere Sortierfelder angibt.

In folgendem Beispiel wird zuerst das Feld dept sortiert und dann das Feld rating. Beide Felder werden aufsteigend sortiert, da der Query-Parameter –sortorder nicht angegeben wurde.

Beispiel:

`http://192.168.123.101/fmi/xml/fmresultset.xml?–db=employees&–lay=performance&–sortfield.1=dept
&–sortfield.2=rating&–findall`

Query-Parameter –sortorder (Sortierfolge)

Gibt die Richtung einer Sortierung an.

Wert: Die Sortierfolge. Gültige Sortierfolgen (wobei <wertelisten-name> der Name einer Werteliste wie Spezial ist):

Schlüsselwort	Äquivalenter Operator in FileMaker Pro
ascend	Sortieren von a bis z, -10 bis 10
descend	Sortieren von z bis a, 10 bis -10
<wertelisten-name>	Sortieren nach der angegebenen Werteliste für das Feld im Layout

Optional für: –find- oder –findall-Query-Befehle

Erfordert: –sortfield (Query-Parameter)

Der Query-Parameter –sortorder kann mit dem Query-Parameter –sortfield verwendet werden, um die Sortierfolge mehrerer Sortierfelder anzugeben. Syntax für die Sortierfolge eines Sortierfelds:

–sortorder.precedence-number=sort-method

Dabei gilt:

- precedence-number im Query-Parameter –sortorder.precedence-number ist eine Zahl von 1 bis 9, die den Query-Parameter –sortfield angibt, auf den sich der Query-Parameter –sortorder bezieht.
- sortier-methode ist eines der Schlüsselwörter in obiger Tabelle für die Festlegung der Sortierfolge, z. B. „ascend“.

Im folgenden Beispiel ist die Sortierfolge für das Sortierfeld mit der höchsten Rangnummer (dept) „ascend“ und für das Sortierfeld mit der zweithöchsten Rangnummer (rating) „descend“. Die precedence-number 2 in –sortorder.2 gibt an, dass der Query-Parameter –sortorder.2=descend sich auf den Query-Parameter –sortfield.2=rating bezieht.

Beispiel:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=performance&-sortfield.1=dept
&-sortorder.1=ascend&-sortfield.2=rating&-sortorder.2=descend&-findall
```

Hinweis Ein Sortierfeld, für das der Query-Parameter `-sortorder` nicht angegeben ist, verwendet standardmäßig aufsteigende Sortierung.

Query-Parameter `-stylehref` (Stil-Href)

Generiert eine Verarbeitungsanweisung für XML-Stylesheets im Ausgabedokument. Dabei setzt er den Wert des href-Attributs (`href=/mystylesheet.css` or `href=/stylesheets/mystylesheet.xml`), damit Sie Client-seitige Cascading Stylesheets (CSS) oder XSLT-Stylesheets mit Ihrem XML-Dokument verwenden können. Der Wert des Parameters `-stylehref` muss ein absoluter Pfad sein. Das Stylesheet kann einen beliebigen Namen haben, aber die Erweiterung muss `.css` oder `.xml` sein. Weitere Informationen finden Sie unter „Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung“ auf Seite 53 Dieser Parameter wird in Verbindung mit dem Parameter `-styletype` verwendet.

Optional für: Alle Query-Befehle

Erfordert: `-styletype`-Parameter

Beispiel (setzt voraus, dass sich `mystylesheet.xml` im Root-Ordner der Web-Server-Software befindet):

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-styletype=text/xml
&-stylehref=/mystylesheet.xml&-findall
```

Query-Parameter `-styletype` (Stilart)

Generiert eine Verarbeitungsanweisung für XML-Stylesheets im Ausgabedokument. Dabei setzt er den Wert des Typattributs (`type=text/css` oder `type=text/xml`), damit Sie Cascading Stylesheets (CSS) oder XSLT-Dokumente mit Ihrem XML-Dokument verwenden können. Weitere Informationen finden Sie unter „Verwenden von Server-seitiger und Client-seitiger Stylesheet-Verarbeitung“ auf Seite 53 Dieser Parameter wird in Verbindung mit dem Parameter `-stylehref` verwendet.

Optional für: Alle Query-Befehle

Erfordert: `-stylehref`-Parameter

Beispiel (setzt voraus, dass sich `mystylesheet.css` im Root-Ordner der Web-Server-Software befindet):

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=departments&-styletype=text/css
&-stylehref=/mystylesheet.css&-findall
```

Query-Parameter `-token.[string]` (Werte zwischen XSLT-Stylesheets übergeben)

Übergibt beliebige benutzerdefinierte Informationen zwischen XSLT-Stylesheets, ohne Sitzungen oder Cookies zu verwenden. Dieser Query-Parameter kann nur für Custom Web Publishing mit XSLT-Abfragen verwendet werden.

***string* in `-token.[string]` ist:** Alphanumerischer String beliebiger Länge, außer Leerzeichen, einschließlich der Ziffern 0-9, Kleinbuchstaben a-z oder Großbuchstaben A-Z.

Benutzerdefinierter Parameterwert ist: Beliebige Zeichenfolge, die URL-kodiert ist.

Optional für: Alle XSLT-Abfragen

Beispiel:

```
http://192.168.123.101/fmi/xsl/template/my_stylesheet.xml?-db=employees&-lay=departments
&-grammar=fmresultset&-token.D100=Active&-findall
```

Weitere Informationen finden Sie unter „Verwendung von Tokens zur Übergabe von Informationen zwischen Stylesheets“ auf Seite 63

Anhang B

Fehlercodes für Custom Web Publishing

Die Web Publishing Engine unterstützt drei Arten von Fehlercodes, die für Custom Web Publishing auftreten können:

- **Datenbank- und Query-String-Fehler.** Die Web Publishing Engine generiert für eine veröffentlichte Datenbank jedes Mal einen Fehlercode, wenn eine XML-Datenabfrage auftritt. Weitere Informationen finden Sie im nächsten Abschnitt, „Fehlercodenummern für FileMaker-Datenbanken“.
- **Web Publishing Engine-Fehler.** Wenn sich die Web Publishing Engine im Entwicklungsmodus befindet, generiert sie eine spezifische Fehlerseite, falls in der Web Publishing Engine selbst ein Fehler auftritt. Im Produktionsmodus wird eine allgemeine Textmeldung angezeigt. Siehe „Fehlercodenummern für die Web Publishing Engine“ auf Seite 120.
- **Fehler in FileMaker XSLT-Erweiterungsfunktionen.** Sie können die Erweiterungsfunktion `fmxml:check_error_status()` in einem XSLT-Stylesheet verwenden, um den Fehlerstatus von Erweiterungsfunktionen bei ihrem Aufruf zu überprüfen. Siehe „Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen“ auf Seite 121.

Eine Liste der neuesten Fehlercodes finden Sie in der FileMaker Knowledge Base (www.filemaker.com/kb/).

Fehlercodenummern für FileMaker-Datenbanken

Die Web Publishing Engine generiert für eine in XML-Format veröffentlichte Datenbank jedes Mal einen Fehlercode, wenn Daten angefordert werden. Diese Art von Fehlercodewert wird am Beginn des XML-Dokuments in das `<error code>`-Element für die `fmresultset`-Grammatik bzw. im `<ERRORCODE>`-Element für die Grammatik `FMPXMLRESULT` oder `FMPDSORERESULT` eingefügt. Der Fehlercode 0 gibt an, dass kein Fehler aufgetreten ist.

Beispiel des Datenbankfehlercodes in der `fmresultset`-Grammatik:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fmresultset PUBLIC "-//FMI//DTD fmresultset//EN" "fmi/xml/fmresultset.dtd">
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset" version="1.0">
  <error code="0"></error>
</fmresultset>
```

Beispiel des Datenbankfehlercodes in der `FMPXMLRESULT`-Grammatik:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMLRESULT PUBLIC "-//FMI//DTD FMPXMLRESULT//EN" "fmi/xml/FMPXMLRESULT.dtd">
<fmpxmlresult xmlns="http://www.filemaker.com/fmpxmlresult">
  <ERRORCODE>0</ERRORCODE>
</fmpxmlresult>
```

Als Entwickler der Custom Web Publishing-Lösung bleibt es Ihnen überlassen, den Wert des `<error code>`- bzw. `<ERRORCODE>`-Elements zu prüfen und geeignete Maßnahmen vorzusehen. Die Web Publishing Engine behandelt keine Datenbankfehler.

Fehlernummer	Beschreibung
-1	Unbekannter Fehler
0	Kein Fehler

Fehlernummer	Beschreibung
1	Aktion durch Benutzer abgebrochen
2	Speicherfehler
3	Befehl nicht verfügbar (z. B. falsches Betriebssystem, falscher Modus etc.)
4	Befehl unbekannt.
5	Befehl ungültig (z. B. ist für den Scriptschritt „Feldwert setzen“ keine Formel angegeben).
6	Datei ist schreibgeschützt.
7	Speicherüberlauf
8	Leeres Ergebnis
9	Ungenügende Berechtigungen
10	Angeforderte Daten fehlen.
11	Name ist nicht gültig.
12	Name existiert bereits.
13	Datei oder Objekt ist in Gebrauch.
14	Außerhalb des gültigen Bereichs
15	Teilen durch null nicht möglich.
16	Operation fehlgeschlagen, Anfrage wiederholen (z. B. eine Benutzeranfrage).
17	Konvertierung von fremdem Zeichensatz in UTF-16 fehlgeschlagen.
18	Client muss Kontoinformationen liefern, um fortzufahren.
19	Zeichenfolge enthält andere Zeichen als A-Z, a-z, 0-9 (ASCII).
20	Befehl oder Vorgang durch ausgelöstes Script abgebrochen.
100	Datei fehlt.
101	Datensatz fehlt.
102	Feld fehlt.
103	Beziehung fehlt.
104	Script fehlt.
105	Layout fehlt.
106	Tabelle fehlt.
107	Index fehlt.
108	Werteliste nicht vorhanden.
109	Berechtigung fehlt.
110	Bezugstabellen fehlen.
111	Feldwiederholung ist ungültig.
112	Fenster fehlt.
113	Funktion fehlt.
114	Dateiverweis fehlt.
115	Menüset fehlt.
116	Layoutobjekt fehlt.

Fehlernummer	Beschreibung
117	Datenquelle fehlt.
130	Dateien sind beschädigt oder fehlen und müssen neu installiert werden.
131	Dateien des Sprachpakets fehlen (z. B. Vorlagendateien).
200	Zugriff auf Datensatz verweigert.
201	Feld kann nicht geändert werden.
202	Zugriff auf Feld verweigert.
203	Keine zu druckenden Datensätze in der Datei bzw. Passwort erlaubt kein Drucken.
204	Kein Zugriff auf Feld(er) in Sortierfolge
205	Benutzer hat keine Zugriffsrechte, um neue Datensätze zu erstellen; Import überschreibt bestehende Daten.
206	Benutzer hat keine Zugriffsrechte, um das Passwort zu ändern, oder die Datei ist schreibgeschützt.
207	Benutzer hat nicht genügend Zugriffsrechte, um das Datenbankschema zu ändern, oder die Datei ist schreibgeschützt.
208	Passwort enthält zu wenige Zeichen.
209	Neues Passwort muss sich vom bestehenden unterscheiden.
210	Benutzerkonto ist inaktiv.
211	Passwort ist abgelaufen.
212	Ungültiges Benutzerkonto und/oder Passwort. Versuchen Sie es erneut.
213	Benutzerkonto und/oder Passwort existieren nicht.
214	Zu viele Anmeldeversuche
215	Administratorrechte können nicht dupliziert werden.
216	Gastkonto kann nicht dupliziert werden.
217	Benutzer hat nicht genügend Zugriffsrechte, um Administratorkonto zu ändern.
300	Datei ist geschützt oder in Gebrauch.
301	Datei ist blockiert durch anderen Anwender.
302	Tabelle ist blockiert durch anderen Anwender.
303	Datenbankschema ist blockiert durch anderen Anwender.
304	Layout ist blockiert durch anderen Anwender.
306	Datensatzänderungs-ID stimmt nicht überein.
400	Suchkriterien sind leer.
401	Kein Datensatz entspricht der Abfrage.
402	Kein Abgleichsfeld für eine Referenz
403	Maximales Datensatzlimit für FileMaker Pro-Demo wird überschritten.
404	Ungültige Sortierfolge
405	Angegebene Datensatzzahl übersteigt die Anzahl der ausschließbaren Datensätze.
406	Ungültige Kriterien für Ersetzen/Neunummerierung
407	Ein oder beide Gruppierfeld(er) fehlen (ungültige Beziehung).
408	Angegebenes Feld hat ein Datenformat, das diesem Befehl nicht entspricht.

Fehlernummer	Beschreibung
409	Ungültige Importfolge
410	Ungültige Exportfolge
412	Falsche Version von FileMaker Pro verwendet, um die Datei wiederherzustellen
413	Angegebenes Feld hat ungeeigneten Feldtyp.
414	Layout kann das Ergebnis nicht anzeigen.
415	Ein oder mehrere erforderliche Bezugsdatensätze sind nicht verfügbar.
416	Primärschlüssel erforderlich für Datenquelltable
417	Die Datenbank ist keine unterstützte Datenquelle.
500	Datumswert entspricht nicht den Überprüfungskriterien.
501	Zeitwert entspricht nicht den Überprüfungskriterien.
502	Zahlenwert entspricht nicht den Überprüfungskriterien.
503	Feldwert entspricht nicht der Bereichsüberprüfung.
504	Feldwert entspricht nicht der Eindeutigkeitsüberprüfung.
505	Feldwert existiert nicht in der Datenbank und entspricht nicht der Existenzüberprüfung.
506	Feldwert entspricht nicht der Überprüfung nach Bestandteil einer Werteliste.
507	Feldwert entspricht nicht der Überprüfung durch Berechnung.
508	Ungültiger Wert wurde in Suchenmodus eingegeben.
509	Feld verlangt gültigen Wert.
510	Bezugswert ist leer oder nicht verfügbar.
511	Der Wert im Feld überschreitet die maximale Anzahl der zulässigen Zeichen.
512	Datensatz wurde bereits durch anderen Anwender geändert.
513	Um einen Datensatz zu erstellen, muss der Datensatz in mindestens einem Feld einen Wert haben.
600	Druckerfehler aufgetreten.
601	Kombination von Kopf- und Fußbereich übersteigt eine Seitenlänge.
602	Datenbereich passt für aktuelle Spalteneinstellung nicht auf eine Seite.
603	Verbindung zum Drucker getrennt.
700	Datei hat falschen Dateityp für Import.
706	EPSF-Datei hat keine Übersichtsgrafik.
707	Grafikfilter nicht vorhanden.
708	Dateiimport nicht möglich bzw. Farbmonitor für Import erforderlich.
709	Import des QuickTime-Films fehlgeschlagen.
710	QuickTime-Dateiverweis konnte nicht aktualisiert werden, da für Datenbankdatei nur Lesezugriff besteht.
711	Importfilter nicht vorhanden.
714	Zugriffsrechte reichen für diesen Befehl nicht aus.
715	Benannter Bereich oder Tabellenblatt von Excel konnte nicht gefunden werden.
716	Eine SQL Anfrage mit DELETE, INSERT oder UPDATE ist nicht für ODBC Import erlaubt.
717	Zum Fortsetzen des Imports bzw. Exports sind nicht genügend XML/XSL-Daten vorhanden.

Fehlernummer	Beschreibung
718	XML-Parsingfehler (von Xerces)
719	Fehler beim Transformieren von XML mit XSL (von Xalan)
720	Fehler beim Export: Das vorgesehene Format unterstützt keine Wiederholfelder.
721	Im Parser oder Transformer ist ein unbekannter Fehler aufgetreten.
722	Daten können nicht in eine Datei importiert werden, die keine Felder hat.
723	Sie sind nicht berechtigt, Datensätze in der Zieltabelle hinzuzufügen oder zu ändern.
724	Sie sind nicht berechtigt, Datensätze in der Zieltabelle hinzuzufügen.
725	Sie sind nicht berechtigt, Datensätze in der Zieltabelle zu ändern.
726	In der Importdatei sind mehr Datensätze vorhanden als in der Zieltabelle. Nicht alle Datensätze werden importiert.
727	In der Zieltabelle sind mehr Datensätze vorhanden als in der Importdatei. Nicht alle Datensätze werden aktualisiert.
729	Fehler beim Import. Datensätze konnten nicht importiert werden.
730	Nicht unterstützte Excel-Version. Konvertieren Sie die Datei in Excel 7.0 (Excel 95), Excel 97, 2000, XP oder 2007 und versuchen Sie es erneut.
731	Die für den Import ausgewählte Datei enthält keine Daten.
732	Diese Datei kann nicht eingefügt werden, weil sie selbst weitere Dateien enthält.
733	Eine Tabelle kann nicht in sich selbst importiert werden.
734	Dieser Dateityp kann nicht als Bild dargestellt werden.
735	Dieser Dateityp kann nicht als Bild dargestellt werden. Er wird eingefügt und dann als Datei dargestellt.
736	Zu viele Daten für den Export in dieses Format. Sie werden abgeschnitten.
737	Die Bento-Tabelle, die Sie importieren, fehlt.
800	Datei konnte auf Datenträger nicht erstellt werden.
801	Temporärdatei konnte auf Systemdatenträger nicht erstellt werden.
802	Datei konnte nicht geöffnet werden. Dieser Fehler kann durch eine oder mehrere der folgenden Möglichkeiten verursacht werden: <ul style="list-style-type: none"> ■ Ungültiger Datenbankname ■ Datei ist in FileMaker Server geschlossen. ■ Ungültige Berechtigung
803	Datei in Einzelbenutzer-Status oder Host nicht vorhanden.
804	Datei konnte in ihrem aktuellen Status nicht mit Nur-Lese-Zugriff geöffnet werden.
805	Datei ist beschädigt; stellen Sie die Datei wieder her.
806	Datei kann mit dieser Version von FileMaker Pro nicht geöffnet werden.
807	Datei ist keine FileMaker Pro-Datei oder ist schwer beschädigt.
808	Datei kann wegen beschädigter Zugriffsrechte nicht geöffnet werden.
809	Datenträger voll
810	Datenträger fixiert
811	Temporärdatei kann nicht als FileMaker Pro-Datei geöffnet werden.
813	Fehler bei Datensatz-Synchronisation im Netzwerk

Fehlernummer	Beschreibung
814	Datei(en) kann (können) nicht geöffnet werden, da die maximale Anzahl an Dateien geöffnet ist.
815	Referenzdatei konnte nicht geöffnet werden.
816	Datei konnte nicht konvertiert werden.
817	Der Bindungsschlüssel der Datei stimmt nicht mit dieser Laufzeitanwendung überein.
819	Lokale Kopie einer remote Datei kann nicht gespeichert werden.
820	Datei wird geschlossen.
821	Host hat die Verbindung getrennt.
822	FMI-Dateien nicht gefunden; fehlende Dateien neu installieren.
823	Datei kann nicht auf Einzelbenutzer gesetzt werden, da Gäste verbunden sind.
824	Datei ist beschädigt oder keine FileMaker-Datei.
825	Datei ist nicht autorisiert, die geschützte Datei zu referenzieren.
900	Allgemeiner Fehler in der Rechtschreibprüfung
901	Standardwörterbuch nicht installiert.
902	Hilfe-System konnte nicht gestartet werden.
903	Befehl kann nicht in einer gemeinsam genutzten Datei verwendet werden.
904	Befehl kann nur in einer Datei verwendet werden, die von FileMaker Pro Server freigegeben wurde.
905	Kein aktives Feld ausgewählt, Befehl kann nur mit aktivem Feld verwendet werden.
906	Die aktuelle Datei wird nicht bereitgestellt. Der Befehl kann nur verwendet werden, wenn die Datei bereitgestellt wird.
920	Initialisierung der Rechtschreibprüfung nicht möglich.
921	Anwenderwörterbuch kann nicht zur Bearbeitung geladen werden.
922	Anwenderwörterbuch existiert nicht.
923	Anwenderwörterbuch ist schreibgeschützt.
951	Ein unerwarteter Fehler ist aufgetreten.
954	Nicht unterstützte XML-Grammatik
955	Kein Datenbankname
956	Maximale Anzahl von Datenbanksitzungen überschritten.
957	Widersprüchliche Befehle
958	Parameter fehlt in Query
959	Custom Web Publishing ist deaktiviert
1200	Generischer Rechenfehler
1201	In dieser Funktion gibt es zu wenige Parameter.
1202	In dieser Funktion gibt es zu viele Parameter.
1203	Unerwartetes Ende der Berechnung
1204	Es wird eine Zahl, eine Textkonstante, ein Feldname bzw. „(“ erwartet.
1205	Kommentar ist nicht mit „*/“ beendet.
1206	Textkonstante muss mit einem Anführungszeichen enden.
1207	Klammer unvollständig

Fehlernummer	Beschreibung
1208	Operator fehlt, Funktion nicht gefunden oder „(“ nicht erwartet.
1209	Name (z. B. Feldname oder Layoutname) fehlt.
1210	Plugin-Funktion wurde bereits registriert.
1211	In dieser Funktion ist die Listennutzung nicht zulässig.
1212	Hier wird ein Operator (z. B. +, -, *) erwartet.
1213	Diese Variable wurde bereits in der SetzeVars-Funktion definiert.
1214	MITTELWERT, ANZAHL, ERWEITERN, HOLEWIEDERHOLFELDWERT, MAX, MIN, NBW, STABW, SUMME und HOLEERGEBNISWERT: Ausdruck gefunden, wo nur ein Feld benötigt wird.
1215	Dieser Parameter ist ein ungültiger Statusfunktionsparameter.
1216	Als erstes Argument einer HOLEERGEBNISWERT-Funktion können nur Statistikfelder angegeben werden.
1217	Gruppierfeld ist ungültig.
1218	Zahl kann nicht berechnet werden.
1219	Ein Feld kann nicht in seiner eigenen Formel benutzt werden.
1220	Feldtyp muss normal oder berechnet sein.
1221	Datentyp muss Zahl, Datum, Zeit oder Zeitstempel sein.
1222	Formel kann nicht gespeichert werden.
1223	Die angegebene Funktion ist noch nicht implementiert.
1224	Die angegebene Funktion existiert nicht.
1225	Die angegebene Funktion wird in diesem Kontext nicht unterstützt.
1400	ODBC-Client-Treiberinitialisierung fehlgeschlagen. Stellen Sie sicher, dass die ODBC-Client-Treiber richtig installiert sind.
1401	Umgebung konnte nicht zugeteilt werden (ODBC).
1402	Umgebung konnte nicht freigegeben werden (ODBC).
1403	Verbindung trennen fehlerhaft (ODBC).
1404	Verbindung konnte nicht zugeteilt werden (ODBC).
1405	Verbindung konnte nicht freigegeben werden (ODBC).
1406	Überprüfung von SQL API fehlgeschlagen (ODBC).
1407	Anweisung konnte nicht zugeteilt werden (ODBC).
1408	Erweiterter Fehler (ODBC)
1409	Erweiterter Fehler (ODBC)
1410	Erweiterter Fehler (ODBC)
1411	Erweiterter Fehler (ODBC)
1412	Erweiterter Fehler (ODBC)
1413	Erweiterter Fehler (ODBC)
1450	Aktion erfordert PHP-Berechtigungserweiterung.
1451	Aktion erfordert, dass die aktuelle Datei remote ist.
1501	SMTP-Authentifizierung fehlgeschlagen.

Fehlernummer	Beschreibung
1502	Verbindung vom SMTP-Server abgelehnt.
1503	SSL-Fehler
1504	Der SMTP-Server erfordert eine verschlüsselte Verbindung.
1505	Die angegebene Authentifizierung wird vom SMTP-Server nicht unterstützt.
1506	E-Mail(s) konnte(n) nicht erfolgreich versendet werden.
1507	Anmeldung am SMTP-Server nicht möglich

Fehlercodenummern für die Web Publishing Engine

Wenn sich die Web Publishing Engine im Entwicklungsmodus befindet, generiert sie eine spezifische Fehlerseite, sobald in der Web Publishing Engine selbst ein Fehler auftritt. Dieser Fehlertyp kann eine Vielzahl von Ursachen haben, z. B. wenn die Web Publishing Engine zu Folgendem nicht in der Lage ist:

- Finden einer angeforderten oder verschachtelten (über <xsl:include>) Stylesheet-Datei
- Parsen einer angeforderten oder verschachtelten Stylesheet-Datei aufgrund eines XML-Fehlers in der Datei
- Generieren eines Stylesheets aus der Datei aufgrund eines XSLT- oder XPath-Fehlers in der Datei
- Verarbeiten der Abfrage, da die XML-Grammatik im CGI nicht korrekt angegeben ist
- Kommunizieren mit dem Web Publishing Core, um XML abzurufen

Wenn die Web Publishing Engine im Entwicklungsmodus läuft, enthält die Fehlerseite für diesen Fehlertyp eine Fehlermeldung und eine eingeklammerte Fehlernummer. Beispiel:



Beispiel für Fehlerseite, wenn sich die Web Publishing Engine im Entwicklungsmodus befindet

Die folgende Liste führt einige der Web Publishing Engine-Fehlercodewerte auf:

Fehlercode-Wert	Beschreibung
QUERY-ER0001	Im Query-Parameter –grammar ist keine XML-Grammatik angegeben.
QUERY-ER0002	„xxx“ ist keine gültige XML-Grammatik für FileMaker XSLT.
FILE-ER0001	Die abgefragte Stylesheet-Datei wurde nicht gefunden.
FILE-ER0002	Die angeforderte Datei wurde nicht gefunden.
UNKNOWN	Ein unerwarteter Fehler ist aufgetreten.
MCS-000 bis MCS-600	Ein unerwarteter Fehler ist aufgetreten.

Fehlercode-Wert	Beschreibung
MCS-601	Die Ressource „x“ konnte nicht geladen werden, weil Ressourcen dieses Typs nicht unterstützt werden: „x“.
MCS-602	Der URL „x“ konnte nicht aufgelöst werden.
MCS-603	Die HTTP-Anfrage für „x“ gab einen Fehler des Typs „x“ zurück.
MCS-604	Die Ressource „x“ konnte nicht geladen werden, weil ein unerwarteter Fehler aufgetreten ist.
MCS-605	Die Ressource „x“ konnte nicht geladen werden, weil der Content-Typ ungültig war.
MCS-606	Die Ressource „x“ konnte nicht geladen werden, weil das Dokument einen XML-Fehler aufweist.
MCS-607	Die Ressource „x“ konnte aufgrund eines Authentifizierungsproblems nicht geladen werden.
MCS-700	Ein unerwarteter Fehler ist aufgetreten.
MCS-800	Ein unerwarteter Fehler ist aufgetreten.

Wenn sich die Web Publishing Engine im Produktionsmodus befindet, wird für Fehler in der Web Publishing Engine die folgende allgemeine Textmeldung auf der Seite `pe_server_error.html` angezeigt:
Bei der Verwendung von FileMaker Custom Web Publishing mit XSLT ist ein unerwarteter Fehler aufgetreten.

Die Standarddatei `pe_server_error.html` enthält die vorangehende Textmeldung in sechs Sprachen.

Als Entwickler können Sie den Text der Fehlerseite `pe_server_error.html` wie für Ihre Lösung erforderlich bearbeiten. Die Datei `pe_server_error.html` befindet sich im Ordner `cwpe` innerhalb des `publishing-engine`-Ordners auf dem Host, auf dem Sie die Web Publishing Engine installiert haben.

Informationen über das Setzen der Web Publishing Engine in den Entwicklungs- oder Produktionsmodus finden Sie in der FileMaker Server Hilfe.

Fehlercodenummern für die FileMaker XSLT-Erweiterungsfunktionen

Die Erweiterungsfunktion `fmxml:check_error_status()` (siehe „Prüfen des Fehlerstatus von Erweiterungsfunktionen“ auf Seite 82) gibt einen der Fehler in der folgenden Tabelle zurück:

Fehlercode-Wert	Beschreibung
-1	Unbekannter Fehler
0	Kein Fehler
Allgemeine Fehler	
10000	Ungültiger Header-Name
10001	Ungültiger HTTP-Statuscode
Sitzungsfehler	
10100	Unbekannter Sitzungsfehler
10101	Angeforderter Sitzungsname bereits in Gebrauch.
10102	Zugriff auf Sitzung nicht möglich – sie existiert evtl. nicht.

Fehlercode-Wert	Beschreibung
10103	Zeitüberschreitung der Sitzung
10104	Angegebenes Sitzungsobjekt existiert nicht.
Nachrichtenfehler	
10200	Unbekannter Nachrichtenfehler
10201	Nachrichten-Formatfehler
10202	SMTP-Feldfehler in Nachricht
10203	Fehler in „An-Feld“ der Nachricht
10204	Fehler in „Von-Feld“ der Nachricht
10205	Fehler in „CC-Feld“ der Nachricht
10206	Fehler in „BCC-Feld“ der Nachricht
10207	Fehler in „Betreff-Feld“ der Nachricht
10208	Fehler in „Reply-To-Feld“ der Nachricht
10209	Nachrichten-Textfehler
10210	Rekursiver Mail-Fehler _ Aufruf von send_email() in E-Mail-XSLT-Stylesheet versucht
10211	SMTP-Authentifizierungsfehler – Anmeldung fehlgeschlagen oder falscher Authentifizierungstyp angegeben
10212	Unzulässige Funktionsverwendung – Aufruf von call set_header(), set_status_code() oder set_cookie() in E-Mail-XSLT-Stylesheet versucht
10213	SMTP-Server ist ungültig oder funktioniert nicht.
Formatierungsfehler	
10300	Unbekannter Formatierungsfehler
10301	Ungültiges Datum-Zeit-Format
10302	Ungültiges Datumsformat
10303	Ungültiges Zeitformat
10304	Ungültiges Tagesformat
10305	Falsch formatierter Datum-Zeit-String
10306	Falsch formatierter Datums-String
10307	Falsch formatierter Zeit-String
10308	Falsch formatierter Tages-String
10309	Nicht unterstützte URL-Textkodierung
10310	Ungültige URL-Textkodierung
10311	Musterfehler in regulärem Ausdruck

Index

A

- Abfragen von Ausschnittfeldern 96
- Abfragen von XML-Daten 37
- Abrufen verfügbarer Scriptnamen 100
- Admin Console für Web Publishing Engine 37
- Administrationskonsole für Web Publishing Engine 29
- Aktivieren von Instant Web Publishing in Datenbank 21
- Angeben der XML-Grammatik 58
- ASCII-Zeichen, in XML-Dokumenten 50
- Attribut für automatische Eingabe 42
- Attribut für vierstellige Jahreszahlen 42
- Ausblenden von Metadaten mit Stylesheets 27
- Ausgabe von Daten mit Stylesheets 27
- Ausgabeseiten
 - Anfängliche Standardkodierungseinstellung 61
 - Ausgabemethode, angeben 62
 - Kodierung, angeben 62
 - <xsl:output>-Element 62
- Ausschnitte
 - Datensätze bearbeiten 96
 - Datensätze hinzufügen 95
 - Datensätze löschen 96
 - Layout 107
 - Sortieren von Datensätzen 107
- Ausschnittfeld-Queries 106, 107
- authenticated base URI parameter 66
- Authentifizierung von Web-Benutzern 21
 - Basis 65
 - Passwort 66
- Authentifizierung Web-Benutzer 21, 65

B

- Basis-URI-Parameter 65
- Befehle, Query s. Query-Strings
- Beispiele für
 - Generierte FMPXMLLAYOUT-Grammatik 49
 - Generierte FMPXMLRESULT-Grammatik 46
 - Generierte fmresultset-Grammatik 44
- Benutzernamen
 - Authentifizierung Web-Benutzer 21, 65
 - Für Custom Web Publishing definieren 21
 - Zugriff auf XML-Dokumente 66
- Berechtigung, Zuweisen für Custom Web

- Publishing 21
- Beschränken von Ausschnittfeldzeilen 107
- break_encode(), Erweiterungsfunktion 75

C

- check_error_status(), Erweiterungsfunktion 82, 121
 - Clientinformationen, über XSLT-Parameter abrufen 65
 - Client-seitige Stylesheets 37, 53
 - compare_date(), Erweiterungsfunktion 78
 - compare_datetime(), Erweiterungsfunktion 79
 - compare_day(), Erweiterungsfunktion 79
 - compare_time(), Erweiterungsfunktion 79
 - Compound-Suche (Query-Befehl) 99
 - Compound-Suche (Query-Parameter) 105
 - contains_checkbox_value(), Erweiterungsfunktion 76
 - convert_datetime(), Erweiterungsfunktion 79
 - Cookies
 - Erweiterungsfunktionen, verwenden 74
 - Sitzungs-ID speichern 68
 - create_session(), Erweiterungsfunktion 69
 - Custom Web Publishing
 - Beschreibung 17, 18
 - Definition 11
 - Erweitertes Zugriffsrecht für 21
 - Gastkonto 22
 - In Datenbank aktivieren 21
 - In Web Publishing Engine aktivieren 22
 - Mit PHP 13
 - Mit XML 13
 - Mit XSLT 13, 28, 55
 - Neue Funktionen in 18
 - Scripts 26
 - Scripts verwenden 24
 - Statische IP-Adresse verwenden 20
 - Übersicht 11
 - Voraussetzungen für das 19
 - Web-Benutzer-Zugr. auf Lösungen 21, 65
 - XML verwenden 35
 - Zugriff von IP-Adressen in Web-Server beschränken 23
 - Custom Web Publishing Engine (CWPE) 28, 36
- ## D
- Element <datasource> 42

Datenbanken, veröffentlichte schützen 22
 Datenbankfehlercodes 41
 Datenbanksitzungen, aktivieren 71, 97
 Datums-Erweiterungsfunktionen, verwenden 78
 Datumsformatstrings 79
 -db (Query-Parameter) 101
 -dbnames (Query-Befehl) 97
 Definieren von Erweiterungsfunktionen 82
 -delete (Query-Befehl) 97
 -delete.related query, Parameter 96
 Document Type Definitions (DTDs) 41, 45
 document(), Funktion 66
 Dokumentation 9
 Dokumente, über document()-Funktion laden 66
 -dup (Query-Befehl) 98

E

-edit (Query-Befehl) 98
 Elektronische Dokumentation 9
 Elemente
 Datenbankfehlercodes 41
 In FMPXMLLAYOUT-Grammatik 46
 In FMPXMLRESULT-Grammatik 45
 In fmresultset-Grammatik 42
 E-Mail-Nachrichten
 Anfängliche Standardkodierungseinstellung 61
 Erweiterungsfunktionen für 71
 -encoding (Query-Parameter) 101
 Entwicklungsmodus, Web Publishing Engine 120
 Erneut anmelden, Script 22
 <error code-> und <ERRORCODE>-Elemente 113
 Erstellen eines neuen Datensatzes 99
 Erweitertes Zugriffsrecht für Custom Web Publishing 21
 Erweiterungsfunktionen für FileMaker XSLT
 Siehe auch fmxslt-Erweiterungsfunktionen
 Exportieren von XML-Daten 35
 Extensible Markup Language (XML) *siehe* XML

F

Fehler
 Datenbankfehlercode-Elemente 41
 Datenbankfehlercodes 113
 Erläuterung von Fehlercodes 113
 Erweiterungsfunktionen,
 Fehlercodenummern 121
 Fehlerstatus von Erweiterungsfunktionen

 prüfen 82, 121
 pe_application_log.txt (Protokolldatei) 90
 pe_server_error.html (Fehlerseite) 121
 Protokolldateien für Web-Server 90
 Web Publishing Engine-
 Fehlercodenummern 120
 Fehlerbehebung
 Custom Web Publishing-Websites 88
 XML-Dokumentzugriff 54
 XSLT-Stylesheets 33
 Feldname (Query-Parameter, nicht für Medienfeld) 102
 Feldnamen, voll qualifizierte Syntax 95
 -field (Query-Parameter, für Medienfeld) 101
 <field-definition> Element 42
 -fieldname.op (Query-Parameter) 103
 FileMaker API for PHP 13
 Definition 13
 FileMaker Pro, Unterschied zu Web Publishing Engine 35
 FileMaker Server
 Dokumentation 9
 Installieren 9
 FileMaker Server Admin
 Siehe Admin-Konsole 22
 FileMaker Server Admin Console 37
 FileMaker Site-Assistent *Siehe* XSLT-Site-Assistent
 FileMaker-spezifische XSLT-Parameter 64
 Filtern von Ausschnittfeldzeilen 106
 Filtern von Daten mit Stylesheets 27
 -find (Query-Befehl) 98
 -findall (Query-Befehl) 98
 -findany (Query-Befehl) 98
 -findquery (Query-Befehl) 99
 FMPDSORESET-Grammatik
 Im Vergleich mit anderen Grammatiken 40
 FMPXMLLAYOUT-Grammatik 35, 46–49
 Im Vergleich mit anderen Grammatiken 40
 FMPXMLRESULT-Grammatik 35, 45–46
 Im Vergleich mit anderen Grammatiken 40
 fmresultset-Grammatik 35, 41–44
 Im Vergleich mit anderen Grammatiken 40
 fmxml (Schlüsselwort zur Aktivierung von XML-Publishing) 21, 37
 fmxml (Schlüsselwort zur Aktivierung von XSLT-Publishing) 21, 29

fmxslt-Erweiterungsfunktionen

fmxslt:break_encode(), Funktion 75
 fmxslt:check_error_status(), Funktion 82, 121
 fmxslt:compare_date(), Funktion 78
 fmxslt:compare_datetime(), Funktion 79
 fmxslt:compare_day(), Funktion 79
 fmxslt:compare_time(), Funktion 79
 fmxslt:contains_checkbox_value(), Funktion 76
 fmxslt:convert_datetime(), Funktion 79
 fmxslt:create_session(), Funktion 69
 fmxslt:get_cookie(), Funktion 74
 fmxslt:get_cookies(), Funktion 74
 fmxslt:get_date(), Funktion 78
 fmxslt:get_datetime(), Funktion 79
 fmxslt:get_day(), Funktion 78
 fmxslt:get_fm_date_format(), Funktion 78
 fmxslt:get_fm_date_format()-Funktion 78
 fmxslt:get_fm_timestamp_format(), Funktion 78
 fmxslt:get_header(), Funktion 73
 fmxslt:get_long_date_format(), Funktion 78
 fmxslt:get_long_day_format(), Funktion 78
 fmxslt:get_long_time_format(), Funktion 78
 fmxslt:get_session_object(), Funktion 70
 fmxslt:get_short_date_format(), Funktion 78
 fmxslt:get_short_day_format(), Funktion 78
 fmxslt:get_short_time_format(), Funktion 78
 fmxslt:get_time(), Funktion 78
 fmxslt:html_encode(), Funktion 75
 fmxslt:invalidate_session(), Funktion 69, 70
 fmxslt:regex_contains(), Funktion 76
 fmxslt:remove_session_object(), Funktion 70
 fmxslt:send_email(), Funktionen 71
 fmxslt:session_encode_url(), Funktion 68, 69
 fmxslt:session_exists(), Funktion 69
 fmxslt:set_cookie(), Funktion 74
 fmxslt:set_header(), Funktion 73
 fmxslt:set_session_object(), Funktion 70
 fmxslt:set_session_timeout(), Funktion 69
 fmxslt:set_status_code(), Funktion 73
 fmxslt:url_decode(), Funktion 76
 fmxslt:url_encode(), Funktion 75

Formatieren von Daten mit Stylesheets 27

Formatstrings, Datum und Zeit 79

G**Gastkonto**

Aktivieren 22

Deaktivieren 22

Mit Custom Web Publishing 22

Generieren einer statischen Seite 62

get_cookie(), Erweiterungsfunktion 74

get_cookies(), Erweiterungsfunktion 74

get_date(), Erweiterungsfunktion 78

get_datetime(), Erweiterungsfunktion 79

get_day(), Erweiterungsfunktion 78

get_fm_date_format(), Erweiterungsfunktion 78

get_fm_time_format(), Erweiterungsfunktion 78

get_fm_timestamp_format(),

Erweiterungsfunktion 78

get_header(), Erweiterungsfunktion 73

get_long_date_format(), Erweiterungsfunktion 78

get_long_day_format(), Erweiterungsfunktion 78

get_long_time_format(), Erweiterungsfunktion 78

get_session_object(), Erweiterungsfunktion 70

get_short_date_format(), Erweiterungsfunktion 78

get_short_day_format(), Erweiterungsfunktion 78

get_short_time_format(), Erweiterungsfunktion 78

get_time(), Erweiterungsfunktion 78

GIF-Dateien, im Web veröffentlichen 24

Globales Attribut 42

–grammar (Query-Parameter) 58, 104

Grammatik, für XSLT empfohlen 58

Grammatiken für XML, Beschreibung 40

H

Header-Funktionen, verwenden 73

Hinweise zur Dokumentation 9, 20

HTML

Formulare für XML-Abfragen 37

XML-Daten umformatieren in 35

html_encode(), Erweiterungsfunktion 75

I

Importieren von XML-Daten 35

Informationen zwischen Stylesheets übergeben 63

Inhaltspufferung, verwenden 68

Installationsdokumentation 9

Instant Web Publishing

Definition 11

Dokumentation 9

Integrieren von Daten mit Stylesheets 27

invalidate_session(), Erweiterungsfunktion 69, 70

ISO-2022-JP-Kodierung 61

ISO-8859-15-Kodierung 61

ISO-8859-1-Kodierung 61

J

- JavaScript
 - Definieren von Erweiterungsfunktionen 82
- JDBC-Dokumentation 9
- JPEG-Dateien, im Web veröffentlichen 24
- jsessionid-Parameter 68

K

- Kodierung
 - Aufträge 61
 - Ausgabe über Element <xsl:output> 62
 - encoding (Query-Parameter) 61, 101
 - Erweiterungsfunktionen zur Stringmanipulation verwenden 75
 - URL 39, 68
 - XML-Daten 40, 50
 - XSLT-Stylesheets 62
- Konten und Zugriffsrechte
 - Für Custom Web Publishing aktivieren 21
 - Gastkonto 22
 - Scripts 24
- Kontrollkästchen, Werte prüfen in 76

L

- Laden zusätzlicher Dokumente 66
- lay (Query-Parameter) 52, 104
- lay.response (Query-Parameter) 52, 104
- Layout bei Abfrage von XML-Daten angeben 52
- Layout wechseln für XML-Antwort 52
- Layoutinformation, in Stylesheet verwenden 67
- Layoutinformationen abrufen 100
- Layoutnamen abrufen 99
- layoutnames (Query-Befehl) 99
- Layouts, für XML-Antwort wechseln 52
- Logs (Ordner) 90
- lop (Query-Parameter) 104
- Löschen von Ausschnittdatensätzen 96

M

- Mail-Nachrichten *siehe* E-Mail-Nachrichten
- max (Query-Parameter) 105
- max-characters, Attribut 42
- max-repeat, Attribut 42

Medienfelder

- Inhalt veröffentlichen 23, 33
- URL-Syntax für Zugriff in XML-Lösungen 38
- URL-Syntax für Zugriff in XSLT-Lösungen 57
- Wie Web-Benutzer auf Daten zugreifen 24
- Element <metadata> 42
- Metadaten, mit Stylesheets ausblenden 27
- method-Attribut, <xsl:output>-Element 62
- MIME (Multipurpose Internet Mail Extensions)-Typen 23
- modid (Query-Parameter) 105

N

- name, Attribut 43
- Namespaces für
 - XML 40
 - XSLT 59
- Neue Funktionen in Custom Web Publishing 18
- new (Query-Befehl) 99
- not-empty, Attribut 42
- numeric-only, Attribut 42
- Nummern für
 - Datenbankfehlercodes 113
 - Erweiterungsfunktions-Fehlercodes 121
 - Web Publishing Engine-Fehlercodes 120

O

- ODBC-Dokumentation 9
- Online-Dokumentation 9
- Operatoren, Vergleich 103

P

- Parameter für XSLT, FileMaker-spezifisch 64
- Parameter, Query *s.* Query-Strings
- Passwort ändern (Script) 22
- Passwörter
 - Authentifizierung Web-Benutzer 21, 65
 - Für Custom Web Publishing definieren 21
 - Kein Anmeldepasswort 22
 - Passwort ändern (Script) 22
 - Zugriff auf XML-Dokumente 66
- PDFs 9
- pe_application_log.txt (Protokolldatei) 90
- pe_internal_access_log.txt (Protokolldatei) 91
- pe_server_error.html (Fehlerseite) 121
- PHP
 - Fehlerbehebung 33
 - Vorteile 14

PHP API für Custom Web Publishing 13
 PHP-Publishing testen 33
 –process (Query-Befehl) 62, 100
 Produktionsmodus, Web Publishing Engine 121
 Programmprotokoll 82, 90
 Protokolldateien 88, 91
 Beschreibung 90
 pe_application_log.txt 90
 pe_internal_access_log.txt 91
 Protokollierung über <xsl:message>-Element 82
 web_server_module_log.txt 91
 Web-Server-Zugriff 90
 <xsl:message>-Element 90
 Pufferung, in Stylesheet verwenden 68

Q

–query (Query-Parameter) 105
 Query-Information, Zugriff in Abfrage 64
 Query-String-Referenz 94
 Query-Strings 50, 58, 93
 Befehle und Parameter 50, 58, 93
 Datensätze in Ausschnitte einfügen 95
 Datensätze in Ausschnitten bearbeiten 96
 Query-String-Referenz 94
 Richtlinien für 94
 Statisch definiert in XSLT-Stylesheets 59
 Variablenfeld, Syntax 97
 Voll qualifizierter Feldname, Syntax 95
 XML 50, 93
 XSLT-Stylesheets, verwenden in 58
 QuickTime-Filme, im Web veröffentlichen 23

R

–recid (Query-Parameter) 106
 regex_contains(), Erweiterungsfunktion 76
 Reguläre Perl-Ausdrücke, Vergleich von Strings 76
 Reihenfolge der XML-Abfrageverarbeitung 53
 <relatedset-definition> Element 43
 –relatedsets.filter (Query-Parameter) 106
 –relatedsets.max (Query-Parameter) 107
 remove_session_object(), Erweiterungsfunktion 70
 result, Attribut 42
 Element <resultset> 43

S

SAT
 Siehe Admin-Konsole 22
 Scalable Vector Graphics (SVG), Umwandeln von

XML-Daten in 35
 Schlüsselwörter zur Aktivierung von Custom Web Publishing 21, 29, 37
 Schützen veröffentlichter Datenbanken 22
 –script (Query-Parameter) 107
 –script.param (Query-Parameter) 108
 –script.prefind (Query-Parameter) 108
 –script.prefind.param (Query-Parameter) 108
 –script.presort (Query-Parameter) 109
 –script.presort.param (Query-Parameter) 109
 –scriptnames (Query-Befehl) 100
 Scripts
 Datenbanksitzungen, aktivieren 71
 Erneut anmelden 22
 Für XML-Abfragen 37
 In Custom Web Publishing 24
 Konten und Zugriffsrechte 24
 Passwort ändern 22
 Tipps und Überlegungen 24
 Trigger 26
 send_email(), Erweiterungsfunktionen 71
 Server-seitige XSLT-Stylesheets 27, 55
 session_encode_url(), Erweiterungsfunktion 68
 session_exists(), Erweiterungsfunktion 69
 set_cookie(), Erweiterungsfunktion 74
 set_header(), Erweiterungsfunktion 73
 set_session_object(), Erweiterungsfunktion 70
 set_session_timeout(), Erweiterungsfunktion 69
 set_status_code(), Erweiterungsfunktion 73
 Shift_JIS-Kodierung 61
 Sicherheit
 Dokumentation 13
 Konten und Passwörter 23
 Richtlinien zum Schützen veröffentlichter Datenbanken 22
 Statisch definierte Query-Strings, verwenden 59
 Zugriff von IP-Adressen beschränken 23
 Site-Assistent
 Beschreibung 18
 Site-Assistent *Siehe* XSLT-Site-Assistent
 Sitzungserweiterungsfunktionen, in Stylesheets verwenden 68, 69
 –skip (Query-Parameter) 109
 –sortfield (Query-Parameter) 110
 –sortorder (Query-Parameter) 110
 Speichern von Informationen in Sitzungen 68, 69
 SSL (Secure Sockets Layer)-Verschlüsselung 23
 Statisch definierte Query-Strings in XSLT-Stylesheets 59

Statisches Veröffentlichen, Definition 11
 Status, in Sitzungen speichern 68, 69
 Strings
 Erweiterungsfunktionen zur Stringmanipulation
 verwenden 75
 Vergleich über reguläre Perl-Ausdrücke 76
 –stylehref (Query-Parameter) 111
 Stylesheets
 Anweisung zur Verarbeitung von XML-
 Stylesheets 53
 Beispiele für Verwendung 27
 Client-seitig 53
 Cookie-Erweiterungsfunktionen 74
 Datums- und Zeitformatstrings 79
 Datums-, Zeit- und Tages-
 Erweiterungsfunktionen 78
 E-Mail-Nachrichten, senden 71
 Entwickeln 55
 Erläuterung 27
 Erstellen mit XSLT-Site-Assistent 30
 Erweiterungsfunktionen zur
 Stringmanipulation 75
 Fehlerstatus von Erweiterungsfunktionen,
 prüfen 82
 –grammar-Parameter für 58
 Header-Funktionen, verwenden 73
 Inhaltspufferung, verwenden 68
 Kodierung von 62
 Kontrollkästchen, Werte prüfen in 76
 Layoutinformation verwenden in 67
 Query-Strings für 58
 Richtlinien für Entwicklung 55
 Server-seitig 27, 55
 Sitzungsfunktion, verwenden 68, 69
 Testen 88, 89
 Vergleich von Strings über reguläre Perl-
 Ausdrücke 76
 Verwenden in Website oder Programm 32
 XSLT, Beschreibung 27
 –styletype (Query-Parameter) 111

T

Tages-Erweiterungsfunktionen, verwenden 78
 Technologietests 33
 Testen
 Websites 88
 XML-Ausgabe 89
 Textkodierung
 Anfängliche Standardeinstellungen 61
 –encoding (Query-Parameter) 61, 101
 Erweiterungsfunktionen zur Stringmanipulation

 verwenden 75
 Für XSLT-Abfragen 61
 Generierte XML-Daten 40
 Kodierungseinstellungen 61
 Standard für Abfragen und Ausgabeseiten 61
 URL 39, 68
 time-of-day, Attribut 42
 –token (Query-Parameter) 63, 111
 Tool für XSLT, Beschreibung 18
 Tools für XSLT, Beschreibung 30
 Trigger 26
 type, Attribut 43

U

Überblick über Schritte für
 XML-Datenzugriff 37
 XSLT Publishing 29
 Übersicht
 Custom Web Publishing 11
 Überwachen von Websites 90
 Umwandeln von Daten mit Stylesheets 27
 Unicode-Zeichen 50
 url_decode(), Erweiterungsfunktion 76
 url_encode(), Erweiterungsfunktion 75
 URL-Syntax für
 Medienobjekte in XML-Lösungen 38
 Medienobjekte in XSLT-Lösungen 57
 XML-Abfragen 38
 XSLT-Stylesheets 56
 URL-Textkodierung 39
 US-ASCII-Kodierung 61
 User-Agent-Header, prüfen 63
 UTF-8 (Unicode Transformation 8 Bit)
 Format 39, 50
 Kodierungseinstellung 61

V

Variablenfelder
 Datenbanksitzungen, aktivieren 71, 97
 Mit Sitzungen verwenden 71, 97
 Syntax 97
 vCards, Umformatieren von XML-Daten in 35
 Verarbeiten von XSLT-Stylesheets 100
 Verarbeitung einer Web Publishing Engine-
 Anforderung 12
 Verfügbare Datenbanklayouts 99
 Verfügbare Scripts 100
 Vergleich von Strings 76

- Vergleichsoperatoren für Felder 103
- Veröffentlichen im Web
 - Datenbankfehlercodes 113
 - Medienfeldobjekte 23, 33
 - Mit XSLT 18, 29, 55
 - QuickTime-Filme 23
 - Schützen von Datenbanken 22
 - Verbindung zum Internet oder Intranet 20
 - Voraussetzungen für das 19
 - XML verwenden 17, 37
- view (Query-Befehl) 100
- Voll qualifizierter Feldname, Syntax 95
- Voraussetzungen für Custom Web Publishing 19
- Vorteile von XML und XSLT 14

W

- Web Publishing Core
 - Illustration 28, 36
 - Interne Zugriffsprotokolle 91
- Web Publishing Engine
 - Admin Console 37
 - Administrationskonsole 29
 - Anforderungsverarbeitung 12
 - Beschreibung 12
 - Entwicklungsmodus 120
 - Erzeugte Fehlercodes 113
 - Produktionsmodus 121
 - Programmprotokoll 90
 - Seiten aus XSLT-Stylesheet generieren 28
 - Vorteile 17
 - XML-Daten generieren 36
 - XML-Dokumente erzeugen 37
- web_server_module_log.txt (Protokolldatei) 91
- Web-Benutzer
 - Medienfelddaten verwenden 24
 - Voraussetzungen für Zugriff auf Custom Web Publishing-Lösungen 19
 - Zugriff auf geschützte Datenbanken 21, 65, 66
- Webbrowser
 - Ausgabe empfangen 12
 - Rolle in XML-Abfragen 36
 - Rolle in XSLT-CWP-Abfragen 28
- Web-Ordner, Medienfeldobjekte kopieren 23
- Web-Server
 - MIME-Typenunterstützung 23
 - Protokolldateien 90
 - Rolle in XML-Abfragen 36
 - Rolle in XSLT-CWP-Abfragen 28

- Websites
 - FileMaker Support-Seiten 9
 - Mit FileMaker Web Publishing Engine erstellen 17
 - Testen 88
 - Überwachen 90
- Wechseln des Layouts für eine XML-Antwort 52
- Werte, prüfen in Kontrollkästchen 76
- wpc_access_log.txt (Datei) 91

X

XML

- Anweisung zur Verarbeitung von XML-Stylesheets 53
- Beschreibung 35
- Client-seitige Stylesheets verwenden 53
- Daten abfragen 37
- Daten filtern 35
- Document Type Definitions (DTDs) 41, 45
- Fehlerbehebung beim Zugriff auf XML-Dokumente 54
- FMPXMLLAYOUT-Grammatik 46
- FMPXMLRESULT-Grammatik 45
- fmresultset-Grammatik 42
 - <field-definition> Element 42
 - <relatedset-definition> Element 43
 - Element <datasource> 42
 - Element <metadata> 42
 - Element <resultset> 43
- Grammatiken, Beschreibung 40
- Im UTF-8-Format kodiert 40, 50
- In Datenbank aktivieren 21
- Namespaces für 40
- Parser 37, 50
- Query-Strings 50, 93
- Reihenfolge der Abfrageverarbeitung 53
- URL-Textkodierung 39
- XML 1.0-Spezifikation 35
- XML-Daten aus Abfragen generieren 36
- Zusammenfassung der Schritte für Zugriff auf XML-Daten 37
- XML Custom Web Publishing 13
- XML-Abfrage
 - Layout angeben 52
- XML-Antwort
 - Layout wechseln 52
- XPath-Anweisungen 63
- <xsl:stylesheet>-Element 59, 63, 64, 89
- <xsl:message>-Element 82
- <xsl:output>-Element 62

<xsl:param name="authenticated-xml-base-uri"/>-Parameter 66
 <xsl:param name="client-ip"/>-Parameter 65
 <xsl:param name="client-password"/>-Parameter 65
 <xsl:param name="client-user-name"/>-Parameter 65
 <xsl:param name="request-query"/>-Parameter 64
 <xsl:param name="xml-base-uri"/>-Parameter 65
 <xsl:param>-Element 64
 <xsl:template>-Element 65, 66, 89
 <xsl:variable>-Element 66
XSLT
 Beispiele für Stylesheets 27
 Beschreibung 27
 Cookie-Erweiterungsfunktionen 74
 Datums- und Zeitformatstrings 79
 Datums-, Zeit- und Tages-Erweiterungsfunktionen 78
 E-Mail-Nachrichten, senden 71
 Entwickeln von XSLT-Stylesheets 55
 Erweiterungsfunktionen für FileMaker 63
 Erweiterungsfunktionen zur Stringmanipulation 75
 Fehlerbehebung in Stylesheets 33
 Fehlerstatus von Erweiterungsfunktionen, prüfen 82
 FileMaker-spezifische XSLT-Parameter 64
 -grammar-Parameter 58
 Header-Funktionen, verwenden 73
 In Datenbank aktivieren 21
 Inhaltspufferung, verwenden 68
 JavaScript-Erweiterungen 82
 Kontrollkästchen, Werte prüfen in 76
 Layoutinformation, verwenden 67
 Namespaces für 59
 Query-String-Referenz 94
 Query-Strings für 58
 Seiten aus XSLT-Stylesheet generieren 28
 Server-seitige Stylesheets 27, 55
 Vergleich von Strings über reguläre Perl-Ausdrücke 76
 Verwenden in Stylesheets in Website oder Programm 32
 XSLT 1.0-Spezifikation 27
 XSLT-CWP-Abfragen 28
 XSLT-Site-Assistent, verwenden 30
 xslt-template-files (Ordner) 29, 32, 67
 Zusammenfassung der Schritte für Publishing 29
XSLT Custom Web Publishing 13
 <?xslt-cwp-buffer buffer-content="true"?>

(Verarbeitungsanweisung) 68
 <?xslt-cwp-query?> (Verarbeitungsanweisung) 56, 59
XSLT-Site-Assistent
 Beschreibung 30
 Generierte Stylesheets, Beschreibung 32
 Starten 31
 Verwenden 31
 Verwendung vorbereiten 30
 xslt-template-files (Ordner) 29, 32, 67

Z

Zeit-Erweiterungsfunktionen, verwenden 78
 Zeitformatstrings 79
 Zugriffsprotokolldateien für Web-Server, Beschreibung 90
 Zugriffsrechte 22
 Zusammenfassung der Schritte für
 XML-Datenzugriff 37
 XSLT Publishing 29