

FileMaker® Server 9

Anpassad webbpublicering
med PHP



©2007, FileMaker, Inc. Med ensamrätt.

FileMaker, Inc.
5201 Patrick Henry Drive
Santa Clara, Kalifornien 95054, USA

FileMaker är ett varumärke som tillhör FileMaker, Inc. och är registrerat i USA och andra länder. ScriptMaker och filmappslogotypen är varumärken som tillhör FileMaker, Inc. Övriga varumärken tillhör sina respektive ägare.

FileMakers dokumentation skyddas av lagen om upphovsrätt och det är därför inte tillåtet att mångfaldiga eller distribuera detta dokument utan FileMakers skriftliga medgivande. Dokumentationen får endast användas tillsammans med ett licensierat exemplar av FileMaker-programmet.

Samtliga personer och företag som förekommer i exempel är fiktiva och eventuella likheter med verkliga personer och företag är fullständigt oavsiktliga. En lista över medverkande visas i dokumentet Tillskrivning som medföljer den här programvaran. Omnämmande av tredjepartsprodukter förekommer endast i informationssyfte och ska inte tolkas som förslag eller rekommendationer. FileMaker, Inc. tar inget ansvar när det gäller prestandan hos dessa produkter.

Mer information finns på webbplatsen www.filemaker.com.

Innehållsförteckning

Om denna handbok	7
Kapitel 1	
<i>Introduktion till anpassad webbpublicering</i>	9
Om webbpubliceringsmotorn	10
Hur en webbpubliceringsförfrågan behandlas	10
Anpassad webbpublicering med PHP	11
Anpassad webbpublicering med XML och XSLT	11
Jämföra PHP med XML och XSLT	12
Anledningar till att välja PHP	12
Anledningar till att välja XML och XSLT	12
Kapitel 2	
<i>Anpassad webbpublicering med PHP</i>	15
Viktiga funktioner i anpassad webbpublicering med PHP	15
Krav för webbpublicering	15
Vad som krävs för att publicera en databas med anpassad webbpublicering	16
Vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning	16
Om att ansluta till Internet eller ett intranät	16
Installera FileMaker API för PHP manuellt	17
Så går du vidare	17
Kapitel 3	
<i>Förbereda databaser för anpassad webbpublicering</i>	19
Aktivera anpassad webbpublicering med PHP för databaser	19
Skapa layouter för anpassad webbpublicering med PHP	19
Skydda publicerade databaser	20
Få åtkomst till en skyddad databas	20
Publicera innehållet i containerfält på webben	21
Hur webbanvändare kan visa containerfältdata	22
FileMaker-manus och anpassad webbpublicering	22
Tips och råd om manus	22
Manusfunktion i anpassade webbpubliceringslösningar	24

Kapitel 4

Översikt över anpassad webbpublicering med PHP	25
Hur webbpubliceringsmotorn fungerar med PHP-lösningar	25
Allmänna steg för anpassad webbpublicering med PHP	25
Använda FileMaker PHP Site Assistant för att skapa PHP-webbplatser	26
Innan du använder PHP Site Assistant	27
Starta PHP Site Assistant	27
Använda webbplatser som har skapats med PHP Site Assistant	28

Kapitel 5

Använda FileMaker API för PHP	29
Mer information	29
Referens för FileMaker API för PHP	29
Introduktion till FileMaker API för PHP	29
Exempel till FileMaker API för PHP	30
Använda FileMaker klass	30
FileMaker klass-objekt	30
FileMaker kommandoobjekt	30
Ansluta till en FileMaker-databas	31
Arbeta med poster	31
Skapa en post	32
Duplicera en post	32
Redigera en post	32
Radera en post	33
Köra FileMaker-manus	33
Hämta en lista på tillgängliga manus	33
Köra ett ScriptMaker-manus	33
Köra ett manus innan ett kommando utförs	33
Köra ett manus innan en resultatuppsättning sorteras	34
Köra ett manus efter resultatuppsättningen har genererats	34
Ordning för att utföra manus	34
Arbeta med FileMaker-layouter	35
Använda portaler	35
Lista de portaler som har definierats på en specifik layout	35
Hämta portalnamn för ett specifikt resultatobjekt	35
Hämta information om portaler för en specifik layout	36
Hämta information för en specifik portal	36
Hämta tabellnamnet för en portal	36
Hämta portalposterna för en specifik post	36
Skapa en ny post i en portal	36
Ta bort en post från en portal	36

Använda värdelistor	37
Hämta namnen på alla värdelistor för en specifik layout	37
Hämta en matris över alla värdelistor för en specifik layout	37
Hämta alternativen för en namngiven värdelista	37
Utföra sökningar	38
Använda kommandot Visa alla	39
Använda kommandot Sök efter alla	39
Använda kommandot Utför sökning	39
Använda kommandot Sammanfattad sökning	39
Behandla posterna i ett sökresultat	41
Filtrera portalrader som returnerats av en sökförfrågan	41
Förkontrollera kommandon, poster och fält	42
Förkontrollera poster i ett kommando	43
Förkontrollera poster	43
Förkontrollera fält	43
Behandla kontrollfel	43
Hantera fel	45
Kapitel 6	
<i>Iscensätta, testa och övervaka en webbplats</i>	47
Iscensätta en anpassad webbpubliceringsplats	47
Testa en anpassad webbpubliceringsplats	47
Övervaka webbplatsen	48
Använda webbserveråtkomst och felloggar	48
Använda webbpubliceringsmotorns programlogg	48
Använda felloggen för webbservern	49
Använda webbpubliceringskärnans interna åtkomstlogg	49
Felsöka webbplatsen	49
Bilaga A	
<i>Felkoder för anpassad webbpublicering med PHP</i>	51
Felkodsnummer för FileMaker-databaser	51
Felkodsnummer för PHP-komponenter	57
<i>Register</i>	59

Förord

Om denna handbok

Den här handboken riktar sig till dig som känner till hur PHP fungerar, som kan utveckla webbplatser och skapa databaser med hjälp av FileMaker Pro[®]. Du bör förstå grunderna i databasdesign i FileMaker Pro, och förstå begreppen fält, relationer, layouter, portaler och containrar. Information om FileMaker Pro finns i FileMaker Pro Hjälp.

Handboken innehåller följande information om anpassad webbpublicering med PHP på FileMaker Server:

- vad som krävs för att du ska kunna utveckla en anpassad webbpubliceringslösning med PHP
- hur man publicerar databaser med PHP
- vad webbanvändare behöver för att få åtkomst till en anpassad webbpubliceringslösning
- hur man använder FileMaker API för PHP för att hämta data från databaser som FileMaker Server är värd för.

Viktigt! Du kan hämta FileMaker-dokumentation i PDF-format från www.filemaker.se/downloads. Uppdateringar till det här dokumentet kan också hämtas från den webbplatsen.

Dokumentationen för FileMaker Server innehåller följande information:

Information om	Finns i
Installera och konfigurera FileMaker Server	<i>FileMaker Server startbok</i> FileMaker Server Hjälp
Direkt webbpublicering	<i>FileMaker Handbok för Direkt webbpublicering</i>
Anpassad webbpublicering med PHP	<i>FileMaker Server Anpassad webbpublicering med PHP</i> (denna bok)
Använda PHP Site Assistant	PHP Site Assistant Hjälp
Anpassad webbpublicering med XML och XSLT	<i>FileMaker Server Anpassad webbpublicering med XML och XSLT</i>
Använda XSLT Site Assistant	XSLT Site Assistant Hjälp
Installera och konfigurera drivrutiner för ODBC och JDBC och använd ODBJ och JDBC	<i>Handbok för FileMaker ODBC och JDBC</i>
Hur FileMaker Server Auto Update kan ladda ner den senaste versionen av ett plugin-program till FileMaker Pro-databasklienter	<i>FileMaker Handboken Uppdatera plugin-program</i>

Kapitel 1

Introduktion till anpassad webbpublicering

Med FileMaker Server kan du publicera din FileMaker-databas på Internet eller ett intranät på följande sätt.

Direkt webbpublicering: Med Direkt webbpublicering publicerar du din databas snabbt och smidigt på webben. Du behöver inte ändra dina databasfiler eller installera ytterligare program – alla som har en kompatibel webbläsare och tillgång till Internet eller ett intranät kan ansluta sig till databasen och visa, redigera, sortera och söka i poster om du ger dem behörighet.

Med Direkt webbpublicering måste värddatorn köra FileMaker Pro eller FileMaker Server. Användargränssnittet liknar FileMaker Pro-programmet. Webbsidor och formulär som webbanvändaren samverkar med är beroende av de layouter och vyer som definierats i FileMaker Pro-databasen. Mer information finns i *FileMaker Handbok för Direkt webbpublicering*.

Statisk publicering: Om du ändrar dina data väldigt sällan eller om du inte vill att användarna ska ha en direktanslutning till din databas kan du använda dig av statisk publicering. Vid statisk publicering exporterar du data från en FileMaker Pro-databas och skapar en webbsida som du kan anpassa ytterligare med HTML. Webbsidan ändras inte när informationen i din databas ändras och användarna öppnar inte din databas. (När du använder Direkt webbpublicering uppdateras data i ett webbfönster varje gång webbläsaren skickar en begäran till FileMaker Server.) Mer information finns i *FileMaker Handbok för Direkt webbpublicering*.

Anpassad webbpublicering: Om du vill ha bättre kontroll över den publicerade databasens utseende och funktionalitet använder du de funktioner för anpassad webbpublicering som finns i FileMaker Server. FileMaker Server, som är värd för de publicerade databaserna, kräver inte att FileMaker Pro är installerat eller igång för att anpassad webbpublicering ska vara tillgängligt.

Med anpassad webbpublicering kan du:

- integrera databasen med en annan webbplats
- avgöra hur användarna ska få tillgång till data
- kontrollera hur data ska visas i webbläsare.

FileMaker Server innehåller två tekniker för anpassad webbpublicering:

- Anpassad webbpublicering med PHP: Använd FileMaker API för PHP, med ett objektorienterat PHP-gränssnitt för FileMaker Pro-databaser, för att integrera dina FileMaker-data i ett PHP-webbprogram. Du kan använda PHP Site Assistant för att skapa en komplett PHP-webbplats, eller koda PHP-webbsidor själv.
- Anpassad webbpublicering med XML och XSLT:
 - Använd XML-datapublicering för att utbyta FileMaker-data med andra webbplatser och program.
 - Använd serverbehandlade XSLT-formatmallar för att integrera eventuella delmängder med FileMaker-data med andra webbplatser, med andra programsystem och med anpassade program. Du kan använda XSLT Site Assistant för att generera XSLT-formatmallar, eller koda formatmallarna själv.

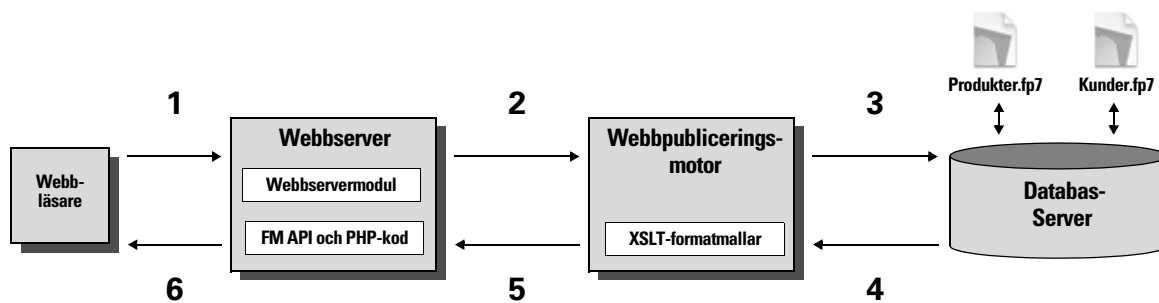
Om webbpubliceringsmotorn

För att stödja Direkt webbpublicering och anpassad webbpublicering använder FileMaker Server en uppsättning programvarukomponenter som kallas för *FileMaker Server Web Publishing Engine*. Webbpubliceringsmotorn hanterar samspelet mellan en webbansvändares webbläsare, din webbserver och FileMaker Server.

Anpassad webbpublicering med XML och XSLT: Webbpubliceringsmotorn fungerar som en XSLT-processor och genererar utdata som HTML, XML eller text (exempelvis vCards (elektroniska visitkort)) till webbservern, som i sin tur skickar dessa utdata till webbläsaren. Webbanvändare får åtkomst till din anpassade webbpubliceringslösning genom att klicka på en HREF-länk eller skriva en URL-adress till webbservern och en FileMaker-frågesträng. URL-adressen kan gå till XML-data eller en XSLT-formatmall. Webbpubliceringsmotorn returnerar de XML-data som angavs i frågesträngen eller resultatet från den angivna XSLT-formatmallen.

Anpassad webbpublicering med PHP: När en webbanvändare ansluter till din anpassade webbpubliceringslösning ansluter PHP på FileMaker Server till webbpubliceringsmotorn och svarar genom FileMaker API för PHP.

Använda FileMaker Servers webbpubliceringsmotor för anpassad webbpublicering



Hur en webbpubliceringsförfrågan behandlas

1. En förfrågan skickas från en webbläsare eller ett program till webbservern.
2. Webbservern vidarebefordrar förfrågan genom FileMakers webbservermodul till webbpubliceringsmotorn.
3. Webbpubliceringsmotorn efterfrågar data från databasen som databasservern är värd för.
4. FileMaker Server skickar efterfrågade FileMaker-data till webbpubliceringsmotorn.

5. Webbpubliceringsmotorn konverterar FileMaker-data för att svara på förfrågan.
 - För PHP-förfrågningar svarar webbpubliceringsmotorn på API-förfrågan.
 - För XML-förfrågningar skickar webbpubliceringsmotorn XML-data direkt till webbservern.
 - För XSLT-förfrågningar använder webbpubliceringsmotorn en XSLT-formatmall som formaterar eller konverterar XML-data och genererar utdata som HTML-sidor, ett XML-dokument eller text till webbservern.
6. Webbservern skickar utdata till den efterfrågande webbläsaren eller programmet.

Viktigt! Säkerheten är mycket viktig när du publicerar information på webben. Läs igenom säkerhetsriktlinjerna i *FileMaker Pro Handbok* som kan hämtas som PDF-fil på www.filemaker.se/downloads.

Anpassad webbpublicering med PHP

FileMaker API för PHP ger ett objektorienterat PHP-gränssnitt för FileMaker-databaser. FileMaker API för PHP gör att man kan få åtkomst till både uppgifter och logik som har lagrats i en FileMaker Pro-databas och publicera dem på Internet, eller exportera dem till andra program. API stöder också komplexa och sammansatta sökkommandon för att extrahera och filtrera data som har lagrats i FileMaker Pro-databaser.

PHP utformades ursprungligen som ett procedurmässigt programmeringsspråk, och har utvecklats som ett objektorienterat webbutvecklingsspråk. PHP ger programspråksfunktionalitet för att skapa praktiskt taget alla typer av logik på en webbsida. Till exempel kan du använda villkorliga logikkonstruktioner för att kontrollera sidgenerering, datarouting eller arbetsflöde. PHP ger även möjlighet till webbplatsadministration och säkerhet.

Dessutom kan du använda FileMaker PHP Site Assistant för att skapa PHP-kod som innehåller alla nödvändiga förutsättningar och funktioner för att få korrekt åtkomst till data i en FileMaker Pro-databas. PHP Site Assistant genererar en flersidig webbplats som gör att webbanvändarna kan söka i en databas, visa en lista över poster, granska poster, sortera poster, lägga till poster, redigera poster, duplicera poster, radera poster och visa en sammanfattningsrapport. FileMaker-utvecklare som har liten erfarenhet av PHP kan använda PHP Site Assistant för att generera en komplett PHP-webbplats. PHP-utvecklare som har liten erfarenhet av FileMaker kan använda PHP Site Assistant för att förstå FileMaker API för PHP-objekt och -metoder.

Anpassad webbpublicering med XML och XSLT

FileMaker anpassad webbpublicering med XML gör att du kan skicka förfrågningar till en FileMaker Pro-databas som FileMaker Server är värd för, och visa, ändra eller manipulera resulterande data. Med en HTTP-förfrågan som innehåller rätt frågekommandon och -parametrar kan du hämta FileMaker-data som ett XML-dokument. Du kan därefter exportera XML-informationen till andra program eller tillämpa en XSLT-formatmall på XML-informationen.

FileMaker anpassad webbpublicering med XSLT gör att du kan omvandla, filtrera eller formatera XML-data för webbläsare eller andra applikationer. Du kan:

- använda en XSLT-formatmall och omvandla informationen mellan FileMaker XML-syntax och en annan XML-syntax för användning i andra program eller andra databaser.

- filtrera informationen genom att välja vilka databasfält som formatmallen ska publicera.
- utforma informationen på en webbsida och ange hur webbanvändaren ska kunna samverka med informationen.

Webbpubliceringsmotorn använder formatmallarna för att hämta data från en FileMaker-databas varje gång en webbanvändare skickar en HTTP-förfrågan och en URL-adress som hänvisar till någon av dina XSLT-formatmallar. Webbpubliceringsmotorn transformerar och formaterar XML-informationen med hjälp av en formatmall och genererar en HTML-sida som webbanvändaren kan arbeta med.

Dessutom är FileMaker Site Assistant ett program som du kan använda när du skapar grundläggande XSLT-formatmallar som kan användas som utgångspunkt för anpassad webbpublicering med XSLT. XSLT Site Assistant skapar formatmallar för sidor som söker igenom databasen, granskar en post i taget, listar posterna i databasen, sorterar poster, lägger till poster, redigerar poster, duplicerar poster, raderar poster och visar en sammanfattningsrapport.

Jämföra PHP med XML och XSLT

Följande avsnitt ger riktlinjer för hur du avgör vilken lösning som passar bäst för din webbplats.

Anledningar till att välja PHP

- PHP är ett mer kraftfullt, objektorienterat procedurmässigt manusspråk och är relativt enkelt att lära sig. Det finns många tillgängliga resurser för utbildning, utveckling och support.
- FileMaker API för PHP gör att man kan få åtkomst till uppgifter och logik som har lagrats i en FileMaker Pro-databas och publicera dem på Internet, eller exportera dem till andra program.
- Med PHP kan man använda villkorslogik för att kontrollera sidkonstruktionen eller flödet.
- PHP ger programspråksfunktionalitet för att skapa många typer av logik på en webbsida.
- PHP är ett av de populäraste webbmanusspråken.
- PHP är ett språk med öppen källkod, som finns på <http://php.net>.
- PHP ger åtkomst till ett stort utbud av komponenter från tredje part som du kan integrera i dina lösningar.

Anledningar till att välja XML och XSLT

- FileMaker XML frågeparametersyntax är utformat för interaktion mellan databaser, vilket underlättar lösningsutveckling.
- XML och XSLT är W3C-standarder.
- XML är ett format som kan läsas av både datorer och människor och som stöder Unicode, vilket gör att data kan kommuniceras på alla skriftliga språk.
- XML passar bra för att presentera poster, listor och trädstrukturerade data.
- Med XSLT kan du omvandla XML-utdata till strukturerade textdokument som RSS, RTF och vCard.
- Du kan använda XSLT för att omvandla XML-utdata från en syntax till en annan.
- Mallar gör det enkelt att tillämpa villkorsformatering på variabla data.

- Du kan använda FMPXMLRESULT-baserade formatmallar för anpassad webbpublicering och för XML-export från FileMaker Pro-databaser.
- FileMaker Server hanterar FileMaker XSLT-formatmallsbehandling, förhindrar obehörig åtkomst till data som kan vara oskyddade med hjälp av XSLT-formatmallar på klientsidan.

Obs! Mer information om anpassad webbpublicering med XSLT finns på *FileMaker Server Anpassad webbpublicering med XSLT*.

Kapitel 2

Anpassad webbpublicering med PHP

Med anpassad webbpublicering med PHP kan du använda programmeringsspråket PHP för att integrera data från FileMaker-databaser i dina egna webbplatslayouter. Med anpassad webbpublicering med PHP får du FileMaker API för PHP, som är en PHP-klass som har skapats av FileMaker som ger åtkomst till databaser som FileMaker Server är värd för. Denna PHP-klass ansluter till FileMaker Servers webbpubliceringsmotor och gör data tillgängliga för din webbservers PHP-motor.

Viktiga funktioner i anpassad webbpublicering med PHP

- Skapa webbapplikationer som använder programmeringsspråket PHP med öppen källkod. Använd FileMaker Servers stödda version av PHP 5, eller använd din egen version av PHP 4.3, PHP 4.4 eller PHP 5. (Om du väljer att använda din egen version av PHP, se "Installera FileMaker API för PHP manuellt" på sidan 17.)
- Använd FileMaker Server som värd för databaser. FileMaker Pro krävs inte för anpassad webbpublicering eftersom FileMaker Server är värd för databaserna.
- Använd PHP Site Assistant för att skapa PHP-koden för en webbplats som ger åtkomst till data i en delad FileMaker-databas. Mer information finns i "Använda FileMaker PHP Site Assistant för att skapa PHP-webbplatser" på sidan 26.
- Skriv PHP-kod som kan skapa, radera, redigera och duplicera poster i en delad FileMaker-databas. Din kod kan utföra kontroll av fält och poster innan ändringar verkställs i den delade databasen.
- Skriv PHP-kod som ger åtkomst till layouter, portaler, värdelistor och relaterade fält. Precis som i FileMaker Pro är åtkomsten till data, layouter och fält baserad på användarkontots inställningar som anges i databasens behörighetsinställningar. Webbpubliceringsmotorn har också flera ytterligare säkerhetsförbättringar. Se "Skydda publicerade databaser" på sidan 20.
- Skriv PHP-kod som utför komplexa manus med flera steg. Cirka 70 manussteg från ScriptMaker™ kan nu utföras på webben. Se "FileMaker-manus och anpassad webbpublicering" på sidan 22.
- Skriv PHP-kod som utför komplexa sökningar.

Krav för webbpublicering

Detta avsnitt förklarar vad som krävs för att utveckla en anpassad webbpubliceringslösning med hjälp av PHP, vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning och vilken påverkan det kan ha på din server att vara värd för en webbpubliceringslösning.

Vad som krävs för att publicera en databas med anpassad webbpublicering

Om du ska publicera databaser med Anpassad webbpublicering med PHP, behöver du följande:

- en FileMaker Server-driftsättning, som innehåller tre komponenter.
 - en webbserver, antingen Microsoft IIS (Windows) eller Apache (Mac OS). FileMakers webbservermodul har installerats på webbservern.
 - FileMaker Web Publishing Engine
 - FileMaker Databaserver
- PHP installerat på webbservern. FileMaker Server kan installera den version av PHP 5 som stöds, och kan även använda din egen version. Det lägsta kravet för PHP är version 4.3. Mer information om PHP finns på <http://php.net>. Den version av PHP som är installerad på webbservern måste stödja cURL-funktioner (client URL library). Information om cURL finns på, finns <http://php.net/curl>.

Viktigt! När du installerar den version av PHP 5 som stöds av FileMaker Server visas det inte i verktyget Mac OS X Server Admin; det ska inte finnas med i listan. Om du använder verktyget Mac OS X Server Admin för att aktivera PHP, inaktiverar du den version av PHP 5 som stöds av FileMaker och aktiverar din egen version av PHP.

- en eller fler FileMaker Pro-databaser på FileMaker Server
- IP-adressen eller domännamnet på värden där webbservern körs
- en webbläsare och åtkomst till webbservern så att du kan utveckla och testa den anpassade webbpubliceringslösningen

Mer information finns i *FileMaker Server startbok*.

Vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning

Webbanvändare som vill ha åtkomst till en anpassad webbpubliceringslösning som använder PHP behöver följande:

- en webbläsare
- åtkomst till Internet eller ett intranät och webbservern
- IP-adressen eller domännamnet på värden där webbservern körs

Om databasen är lösenordsskyddad måste webbanvändare även uppge ett användarnamn och ett lösenord för ett databaskonto

Om att ansluta till Internet eller ett intranät

När du publicerar databaser på Internet eller ett intranät måste värddatorn köra FileMaker Server och de databaser du vill dela måste köras och vara tillgängliga. Dessutom gäller följande:

- Publicera din databas på en dator med fast Internet- eller intranätanslutning. Det går att publicera databaser utan att ha en fast anslutning, men dessa blir bara tillgängliga för webbanvändarna när din dator är ansluten till Internet eller ett intranät.
- Värddatorn för den webbserver som är en del av FileMaker Server-driftsättningen måste ha en dedicerad statisk (permanent) IP-adress eller ett domännamn. Om du ansluter till Internet via en Internetleverantör (ISP) kan din IP-adress vara dynamiskt allokerad (dvs. olika varje gång du ansluter). Med en dynamisk IP-adress blir det svårare för webbanvändarna att hitta dina databaser. Om du inte är säker på vilken typ av anslutning du ska välja kontaktar du din Internetleverantör eller nätverksadministratör.

Installera FileMaker API för PHP manuellt

När du installerar FileMaker Server har du möjlighet att installera den version av PHP som stöds av FileMaker (PHP 5). Om du redan har en PHP-motor installerad och konfigurerad och du endast vill lägga till FileMaker API för PHP kan du installera FileMaker API för PHP-klassen manuellt för att göra den tillgänglig för dina PHP-manus.

Obs! FileMaker Server har testats med PHP-versionerna 4.3.11, PHP 4.4.1 och PHP 5.1.2. Om du inte använder någon av dessa versioner av PHP, bör du tänka på att uppgradera till den version av PHP som stöds av FileMaker.

Att göra FileMaker API för PHP tillgängligt för dina PHP-manus

När du installerade FileMaker Server följde FileMaker API för PHP-paketet med som en .zip-fil på följande plats:

- För IIS (Windows):
`<enhet>\Program\FileMaker\FileMaker Server\Web Publishing\FM_API_for_PHP_Standalone.zip`
 där <enhet> är den enhet där webbserverkomponenten för din FileMaker Server-driftsättning finns.

- För Apache (Mac OS): `/Bibliotek/FileMaker Server/Web Publishing/FM_API_for_PHP_Standalone.zip`

Filen `FM_API_for_PHP_Standalone.zip` innehåller filen `FileMaker.php` och mappen `FileMaker`. Packa upp filen och kopiera filen `FileMaker.php` och mappen `FileMaker` till någon av dessa platser:

- webbserverns rotkatalog där dina PHP-manus finns.
 - För IIS (Windows): `<enhet>\inetpub\wwwroot` där <enhet> är den enhet där webbpubliceringsmotorn till din FileMaker Server-driftsättning finns.
 - För Apache (Mac OS): `/Bibliotek/WebServer/Documents`
- en av `include_path`-katalogerna i din PHP-installation. Standardplatsen för Mac OS X är `/usr/lib/php`.

Så går du vidare

Här följer några förslag för att hjälpa dig komma igång med att utveckla anpassade webbpubliceringslösningar:

- Använd FileMaker Server Admin Console för att aktivera anpassad webbpublicering. Se FileMaker Server Hjälpsida och *FileMaker Server Startbok*.
- I FileMaker Pro öppnar du varje FileMaker-databas som du vill publicera och ser till att databasen har rätt utökad behörighet aktiverad för anpassad webbpublicering. Se "Aktivera anpassad webbpublicering med PHP för databaser" på sidan 19.
- Mer information om hur man får åtkomst till data i FileMaker-databaser med FileMaker API för PHP finns i kapitel 5, "Använda FileMaker API för PHP".

Kapitel 3

Förbereda databaser för anpassad webbpublicering

Innan du kan använda anpassad webbpublicering med en databas måste du förbereda databasen och skydda den från obehörig åtkomst.

Aktivera anpassad webbpublicering med PHP för databaser

Du måste aktivera anpassad webbpublicering med PHP i varje databas som du vill publicera. Annars kan webbanvändare inte använda anpassad webbpublicering för att få åtkomst till databasen, även om en FileMaker Server som är konfigurerad för att stödja en webbpubliceringsmotor är värd för den.

Så här aktiverar du anpassad webbpublicering för en databas:

1. Starta FileMaker Pro och öppna databasen som du vill publicera med hjälp av ett konto som har behörighetsuppsättningen Full åtkomst eller Hantera utökad behörighet.
2. Tilldela utökade fmpHP-behörigheter till en eller flera behörighetsuppsättningar för att möjliggöra anpassad webbpublicering med PHP.
3. Tilldela behörighetsuppsättningarna med utökad behörighet för anpassad webbpublicering till lämpliga konton (till exempel kontona Admin och Gäst).

Viktigt! När du definierar kontonamn och lösenord för lösningar med anpassad webbpublicering använder du utskriftbara ASCII-tecken, till exempel, a-z, A-Z och 0-9. För att få säkrare kontonamn och lösenord ska du ta med några icke alfanumeriska tecken, som ett utropstecken (!) eller procenttecken (%). Kolon (:) är inte tillåtet. Information om hur du gör kontoinställningar finns i FileMaker Pro HjälP.

4. När du använder FileMaker Server Admin Console ska du kontrollera att värden för databasen är korrekt konfigurerad, och att den är tillgänglig för FileMaker Server. Mer instruktioner finns i FileMaker Server HjälP.

Obs! Eftersom anpassad webbpublicering med PHP inte använder beständiga databassessioner, kan referenser till en extern ODBC-datakälla i FileMaker Pro-relationsdiagram begränsa de funktioner som är tillgängliga i din PHP-lösning. Om din databas har åtkomst till data från en extern SQL-datakälla kan det hända att du inte kan uppdatera data i den externa tabellens poster.

Skapa layouter för anpassad webbpublicering med PHP

Anpassad webbpublicering med PHP ger inte direkt tabellåtkomst till data i en FileMaker Pro-databas, utan använder de layouter som har definierats i databasen. Trots att det inte är något krav att skapa en unik layout för anpassad webbpublicering med PHP kan det av flera skäl vara praktiskt att skapa en layout speciellt för en PHP-lösning:

- Du förbättrar prestandan genom att skapa en layout som är begränsad till de fält, etiketter och portaler som du behöver ta med i PHP-lösningen.
- Du förenklar din PHP-kod genom att utföra mindre databehandling eftersom posterna har färre fält.
- Du separerar utformningen av gränssnittet från dina data, så att du kan skraddarsy gränssnittet för webbanvändarna.

Skydda publicerade databaser

Med anpassad webbpublicering med PHP kan du begränsa åtkomsten till dina publicerade databaser. Du kan använda dessa metoder:

- Kräv lösenord för databaskonton som används för anpassad webbpublicering med PHP.
- Aktivera bara den utökade behörigheten anpassad webbpublicering med PHP för de behörighetsuppsättningar som du vill ge åtkomst.
- Inaktivera anpassad webbpublicering med PHP för en specifik databas genom att avmarkera utökad fmpHP-behörighet för alla behörighetsuppsättningar i den databasen. Se FileMaker Pro Hjälp.
- Använd FileMaker Server Admin Console för att aktivera eller inaktivera anpassad webbpublicering i alla anpassade webbpubliceringslösningar i webbpubliceringsmotorn. Se *FileMaker Server startbok* och FileMaker Server Hjälp.
- Konfigurera webbservern för att begränsa vilka IP-adresser som ska kunna använda databasen via webbpubliceringsmotorn. Du kan exempelvis ange att endast webbanvändare från IP-adressen 192.168.100.101 ska ha tillgång till databaserna. Information om hur du begränsar åtkomst från IP-adresser finns i dokumentationen till webbservern.
- Använd SSL-kryptering (Secure Sockets Layer) för kommunikationen mellan webbservern och webbanvändarnas webbläsare. Vid SSL-kryptering konverteras informationen som utväxlas mellan webbservrar och klienter till obegriplig information med matematiska formler, så kallade *chiffer*. Dessa chiffer används när informationen konverteras tillbaka till begripliga data med hjälp av krypteringsnycklar. Information om hur du aktiverar och konfigurerar SSL finns i dokumentationen till webbservern.

Mer information om hur du skyddar din databas finns i *FileMaker Pro Handbok*, som finns som PDF-fil på www.filemaker.se/downloads.

Få åtkomst till en skyddad databas

När en webbanvändare får åtkomst till en databas med en PHP-lösning måste PHP-koden uppge identifieringsuppgifterna till databasen med FileMaker API för PHP. Om gästkontot för databasen är inaktiverat, eller inte har utökad fmpHP-behörighet aktiverad, returnerar FileMaker API för PHP ett felmeddelande och din PHP-kod måste uppge inloggningsuppgifter för användaren.

I introduktionen till FileMaker API för PHP finns ett exempel som visar hur man använder metoden `setProperty()` för att ställa in användarnamn och lösenord för en skyddad databas. Se ”Introduktion till FileMaker API för PHP” på sidan 29.

PHP Site Assistant har två alternativ för att få åtkomst till skyddade databaser:

- PHP-koden kan uppmana webbanvändaren att identifiera sig när användaren ansluter till webbplatsen.
- PHP-koden kan lagra databasens kontonamn och lösenord i webbplatsens PHP-filer.

Mer information finns i PHP Site Assistant Hjälp.

I följande lista sammanfattas processen som sätts igång när man använder anpassad webbpublicering för att få åtkomst till en databas:

- Om inget lösenord har tilldelats för ett konto med anpassad webbpublicering aktiverat, måste PHP-lösningen bara uppge kontonamnet.

- Om gästkontot är inaktiverat måste PHP-lösningen uppge ett kontonamn och ett lösenord. PHP-lösningen kan antingen uppmana webbanvändaren att ange kontonamnet och lösenordet, eller så kan det lagra kontonamnet och lösenordet i PHP-koden. Kontonamnet måste ha den utökade behörigheten `fmphp` aktiverad.
- Om gästkontot är aktiverat och har den utökade behörigheten `fmphp` aktiverad:
 - PHP-lösningen behöver inte uppmana webbanvändarna att ange kontonamn och lösenord när en fil öppnas. Alla webbanvändare loggas automatiskt in med gästkontot och får den behörighet som hör till gästkontot.
 - Standardbehörigheten för gästkonton ger endast läsbehörighet. Du kan ändra standardbehörigheten, även den utökade behörigheten, för detta konto. Se FileMaker Pro Hjälp.
- PHP-lösninge kan använda manussteget Logga in igen för att tillåta användare att logga in med ett annat konto (till exempel för att växla från gästkontot till ett konto med större behörighet). Se FileMaker Pro Hjälp. Eftersom PHP-anslutningar inte använder bestående databassessioner måste PHP-lösningen dock lagra kontonamnet och lösenordet för att använda dem för varje kommande förfrågan.

Obs! Som standardinställning kan webbanvändare inte ändra sina egna kontolösenord från en webbläsare. Denna funktion kan aktiveras för en databas med hjälp av manussteget Ändra lösenord, som gör att webbanvändarna kan ändra sina lösenord från webbläsaren. Se FileMaker Pro Hjälp.

Publicera innehållet i containerfält på webben

Innehållet i ett containerfält, till exempel en bildfil, kan lagras i en FileMaker-databas eller som en filreferens med en relativ sökväg.

Så här använder man innehållet i containerfält i en PHP-lösning:

- Använd korrekta HTML-taggar för att ange vilken typ av webbkompatibelt objekt som containerfältet innehåller, och skapa en URL-sträng som anger sökvägen till HTML-taggens källattribut.

```
<IMG src="img.php?url=<?php echo urlencode($record->getField('Cover Image')); ?>">
```

- Använd FileMaker API för PHP för att definiera databasobjektet med rätt identifieringsuppgifter (kontonamn och lösenord), och använd sedan metoden `getContainerData()` för att hämta containerfältdata.

```
$fm = & new FileMaker();
$fm->setProperty('database', $databaseName);
$fm->setProperty('username', $userName);
$fm->setProperty('password', $passWord);
echo $fm->getContainerData($_GET['-url']);
```

I introduktionen till FileMaker API för PHP finns flera exempel på hur man använder containerfält. Se ”Introduktion till FileMaker API för PHP” på sidan 29.

Om ett containerfält lagrar en filreferens, måste du även följa dessa steg för att publicera de refererade filerna med hjälp av webbpubliceringsmotorn:

1. Lagra containerobjektfilerna i webbappen inuti FileMaker Pro-mappen.
2. Starta FileMaker Pro, sätt in objekten i containerfältet och välj **Spara endast länk till filen**.

3. Kopiera eller flytta objektfilerna som refereras till i webbmappen till samma relativa sökväg i rotmappen i webbserverprogrammet.

- För IIS (Windows): <enhet>\inetpub\wwwroot där <enhet> är den enhet där webbpubliceringsmotorn till din FileMaker Server-driftsättning finns.
- För Apache (Mac OS): /Bibliotek/WebServer/Documents

Obs!

- För att containerobjekt ska kunna lagras som filreferenser måste webbservern kunna hantera MIME-typerna (Multipurpose Internet Mail Extensions) för de typer av filer du vill använda, till exempel filmer. Webbservern fastställer stöd för de aktuella MIME-typer som har registrerats för Internet. Webbpubliceringsmotorn ändrar inte en webbserverns MIME-stöd. Mer information finns i dokumentationen till webbservern.
- Alla QuickTime-filmer som lagras i containerfält lagras efter referens.

Hur webbanvändare kan visa containerfältdata

När du publicerar en databas med webbpubliceringsmotorn gäller följande begränsningar för containerfältdata:

- Webbanvändare kan inte spela upp ljud eller visa OLE-objekt i ett containerfält. I stället visas en bild.
- Webbanvändare kan inte ändra eller lägga till innehåll i ett containerfält. Webbanvändare kan inte använda containerfält för att ladda upp data till databasen.
- Om din databas innehåller bilder i andra format än GIF eller JPEG skapar webbpubliceringsmotorn en tillfällig JPEG-bild när en webbläsare efterfrågar bilddata.
- Webbpubliceringsmotorn stöder inte direktuppspelning av filmfiler. Webbanvändarna måste hämta hela filmfilen innan de kan visa filmen.

FileMaker-manus och anpassad webbpublicering

Funktionen ScriptMaker i FileMaker Pro kan automatisera uppgifter som utförs ofta eller kombinera flera uppgifter. Genom att använda FileMaker-manus tillsammans med anpassad webbpublicering kan webbanvändare utföra en serie uppgifter. Med FileMaker-manus kan man även utföra uppgifter som inte stöds på något annat sätt, till exempel att använda manussteget Ändra lösenord för att ge webbanvändarna möjlighet att ändra lösenord från en webbläsare.

FileMaker stöder cirka 70 manussteg för anpassad webbpublicering. Om du vill se vilka manussteg som inte stöds markerar du kryssrutan Ange webbkompatibilitet i fönstret Ändra manus i FileMaker Pro. De manus som är nedtonade är inte webbkompatibla. Information om hur du skapar manus finns i FileMaker Pro Hjälp.

Tips och råd om manus

Många manussteg fungerar på samma sätt på webben men det finns några som fungerar annorlunda. Se ”Manusfunktion i anpassade webbpubliceringslösningar” på sidan 24. Testa alla manus som ska användas från en webbläsare innan du delar din databas. Logga in med olika användarkonton så att du är säker på att de fungerar för alla klienter.

Här följer några tips och råd:

- Använd konton och behörighet när du vill styra vilka manus en webbanvändare kan utföra. Kontrollera att manusen endast innehåller webbkompatibla manussteg och ge bara tillgång till manus som ska användas från en webbläsare.
- Var också uppmärksam på eventuella biverkningar av manus som utför kombinationer av steg som styrs via behörigheten. Om ett manus exempelvis innehåller steget Radera poster och en webbanvändare inte loggar in med ett konto som ger behörighet att radera poster, utförs inte manussteget Radera poster. Manuset kan dock fortsätta att utföras, vilket kan ge oväntade resultat.
- I fönstret Redigera manus i ScriptMaker kan du välja Kör manus med fullständig behörighet om du vill låta manuset utföra uppgifter som du inte vill låta den enskilda användaren utföra. Du kan till exempel hindra användare från att radera poster genom att begränsa deras konton och behörighet, men ändå tillåta dem att utföra ett manus som raderar vissa typer av poster under vissa förutsättningar som anges i manuset.
- Om ett manus innehåller steg som inte stöds, exempelvis steg som inte är webbkompatibla, kan du använda manussteget Tillåt användaren att avbryta för att avgöra hur efterföljande steg ska hanteras.
 - Om manussteget Tillåt användaren att avbryta är aktiverat (på), stoppas manuset vid manussteg som inte stöds.
 - Om Tillåt användaren att avbryta är avstängt kommer manussteg som inte stöds att hoppas över och manuset att fortsätta.
 - Om det här manussteget inte ingår utförs manus som om funktionen vore aktiverad, så att manus stoppas vid manussteg som inte stöds.
- Vissa manus som fungerar med ett steg från en FileMaker Pro-klient kan kräva ett ytterligare steg, Verkställ post/sökpost, för att spara data på värddatorn. Eftersom webbanvändarna inte har någon direktanslutning till värddatorn får de inte veta när data ändras. Funktioner som till exempel villkorliga värdelistor är inte lika smidiga för webbanvändare eftersom data måste sparas på värddatorn innan effekterna märks i fältet med värdelistan.
- Manus som förändrar data bör innehålla steget Verkställ post/sökpost, eftersom vissa dataändringar inte syns i webbläsaren förrän data sparas eller ”skickas” till servern. Detta gäller flera manussteg som Klipp ut, Kopiera, Klistra in, o.s.v. Många enkelstegsåtgärder bör konverteras till manus och innehålla steget Verkställ post/sökpost. När du skapar manus som ska utföras från en webbläsare ska du inkludera steget Verkställ post/sökpost sist i manuset, så att du är säker på att alla ändringar verkställs.
- Du kan skapa villkorsstyrda manus enligt typen av klient med funktionen Get(Programversion). Om det returnerade värdet innehåller ”Web Publishing Engine” vet du att den aktuella användaren använder din databas via anpassad webbpublicering. För vidare information om dessa funktioner, se FileMaker Pro Hjälp.
- När du har konverterat filerna öppnar du varje manus som webbanvändarna kan tänkas köra och aktiverar Ange webbkompatibilitet för att kontrollera att manusen kan utföras korrekt med anpassad webbpublicering.

Manusfunktion i anpassade webbpubliceringslösningar

Följande manussteg fungerar annorlunda på webben än i FileMaker Pro. Information om samtliga manussteg finns i FileMaker Pro Hjälp.

Manussteg	Funktionssätt i egna webbpubliceringslösningar
Utför manus	Manus kan inte utföras i andra filer, såvida inte filerna ligger på FileMaker Server och Anpassad webbpublicering är aktiverad i de andra filerna.
Avsluta programmet	Loggar ut webbanvändare, stänger alla fönster men avslutar inte webbläsaren.
Tillåt användaren att avbryta	Bestämmer hur manussteg som inte stöds ska hanteras. Välj På om manuset ska stoppas vid steg som inte stöds eller Av om sådana steg ska hoppas över. Närmare information finns i "Tips och råd om manus" på sidan 22. Obs! Webbanvändare kan inte avbryta manus i Anpassad webbpublicering, men detta alternativ gör att manussteg som inte stöds förhindrar att manuset fortsätter.
Manuell felhantering	Detta är alltid aktiverat i anpassad webbpublicering. Webbanvändarna kan inte avbryta manus i Anpassad webbpublicering.
Paus/fortsätt manus	Trots att dessa manussteg stöds i Anpassad webbpublicering, bör du undvika att använda dem. När ett paussteg körs, görs en paus i manuset. Det är bara manus som innehåller ett fortsätt-manussteg som kan återuppta körningen. Om manuset förblir i pausläge tills sessionen avslutas, slutförs inte manuset.
Sortera poster	Du måste spara en viss sorteringsordning med manussteget om det ska utföras i Anpassad webbpublicering.
Öppna URL	Detta manussteg har ingen effekt i en anpassad webbpubliceringslösning.
Gå till fält	Du kan inte använda Gå till fält för att aktivera ett visst fält i webbläsaren, men du kan använda manussteget tillsammans med andra manussteg för att utföra uppgifter. Du kan till exempel gå till ett fält, kopiera innehållet, gå till ett annat fält och klistra in värdet. Se till att spara posten med manussteget Verkställ post om du vill se resultatet i webbläsaren.
Verkställ Post/sökpost	Skickar posten till databasen.

Kapitel 4

Översikt över anpassad webbpublicering med PHP

Med FileMaker API för PHP kan du integrera data från FileMaker Pro-databaser med PHP-lösningar. I detta kapitel beskrivs hur PHP fungerar tillsammans med FileMaker Servers anpassade webbpubliceringsmotor och du får en introduktion av ett verktyg som hjälper dig att komma igång med att skapa PHP-lösningar: FileMaker PHP Site Assistant. Mer detaljerad information om FileMaker API för PHP finns i kapitel 5, ”Använda FileMaker API för PHP”.

Hur webbpubliceringsmotorn fungerar med PHP-lösningar

FileMaker Server består av tre komponenter: en webbserver, webbpubliceringsmotorn och databasservern. (Dessa komponenter kan köras på en dator, två datorer eller tre datorer. Se *FileMaker Server startbok* för mer information.) FileMaker Server är värd för PHP-lösningen när du placerar PHP-filerna på webbservern där PHP-motorn är installerad.

- När en webbanvändare öppnar en PHP-lösning vidarebefordrar webbservern förfrågan till PHP-motorn, som behandlar PHP-koden.
- Om PHP-koden innehåller anrop till FileMaker API för PHP tolkas de anropen och skickas som förfrågningar till webbpubliceringsmotorn.
- Webbpubliceringsmotorn efterfrågar data från databaser som databasservern är värd för.
- Databasservern skickar efterfrågade data till webbpubliceringsmotorn.
- Webbpubliceringsmotorn skickar data till PHP-motorn på webbservern som svar på API-anropet.
- PHP-lösningen behandlar informationen och visar den för webbanvändaren.

Allmänna steg för anpassad webbpublicering med PHP

Här följer en sammanfattning över de steg du måste utföra när du vill använda anpassad webbpublicering med PHP:

1. Kontrollera att PHP-publicering är aktiverat i Admin Console. Se *FileMaker Server startbok*.
2. I FileMaker Pro öppnar du varje FileMaker-databas som du vill publicera och ser till att databasen har den utökade behörigheten `fmphp` aktiverad för anpassad webbpublicering med PHP. Se kapitel 3, ”Förbereda databaser för anpassad webbpublicering”.

Obs! Se till att använda motsvarande FileMaker-databasbehörigheter när du utvecklar PHP-lösningar som ska användas av slutanvändaren. I annat fall kanske du har tillgång till layouter och funktioner i FileMaker-databasen som inte är tillgängliga för slutanvändaren, vilket ger inkonsekventa funktioner.

3. Använd PHP-redigeringsverktygen för att skapa din PHP-lösning.

Du kan använda FileMaker PHP Site Assistant för att skapa den grundläggande PHP-koden för din webbplats. Använd denna genererade kod utan att modifiera den, eller använd den som ett ramverk för att utveckla en mer avancerad webbplats. Den PHP-kod som genererats av PHP Site Assistant innehåller alla nödvändiga förutsättningar och funktioner för att på ett korrekt sätt få åtkomst till data i en FileMaker Pro-databas. Se ”Använda FileMaker PHP Site Assistant för att skapa PHP-webbplatser” på sidan 26.

4. Kopiera eller flytta din webbplats katalogstruktur och dina filer till webbserverns rotkatalog.
 - För IIS (Windows): <enhet>\inetpub\wwwroot där <enhet> är den enhet där webbpubliceringsmotorn till din FileMaker Server-driftsättning finns.
 - För Apache (Mac OS): /Bibliotek/WebServer/Documents
5. Om ett containerfält i en databas lagrar en filreferens istället för en verklig fil måste det hänvisade containerobjektet lagras i FileMaker Pro-katalogen när posten skapas eller redigeras. Du måste kopiera eller flytta objektet till en katalog med samma relativa position i rotkatalogen på webbserverprogramvaran.

Se ”Publicera innehållet i containerfält på webben” på sidan 21.
6. Se till att säkerhetsfunktionerna för platsen eller programmet fungerar.
7. Testa din webbplats genom att använda samma konton och behörigheter som har definierats för webbanvändare.
8. Informera om webbplatsen och gör den tillgänglig. Den URL som webbanvändarna anger har detta format:

http://<server>/<webbplats_sökväg>

- <server> är den maskin där FileMaker Server finns
- <webbplats_sökväg> är den relativa sökväg till startsidan på din webbplats, som avgörs av den katalogstruktur som du använde i steg 4 ovan.

Till exempel, om din webbserver är 192.168.123.101 och din webbplats startsida ligger på webbservern på c:\inetpub\wwwroot\customers\index.php, ska webbanvändaren ange denna URL:

http://192.168.123.101/customers/index.php

Obs! PHP 4 och PHP 5 använder Latin-1-kodning (ISO-8859-1). FileMaker Server returnerar Unicode-data (UTF-8). Använd FileMaker Server Admin Console för att ange standardteckenkodningen för din webbplats. För PHP-webbplatser kan du ange antingen UTF-8 eller ISO-8859-1. Ange samma inställning för charset-attributet i avsnittet <HEAD> för dina PHP-filer på webbplatsen.

Använda FileMaker PHP Site Assistant för att skapa PHP-webbplatser

PHP Site Assistant kan hjälpa dig att skapa en grundläggande PHP-baserad webbplats för att få åtkomst till, manipulera och visa data i en FileMaker Pro-databas som FileMaker Server är värd för. PHP Site Assistant hjälper dig att utforma och konfigurera din webbplats, och skapar PHP-kod baserad på dina indata. PHP Site Assistant innehåller förkonfigurerade teman för att formatera din webbplats.

När din webbplats PHP-kod har skapats kan du anpassa den med hjälp av vanlig text eller ett PHP-redigeringsverktyg. Du kan även använda PHP Site Assistant för att bli bekant med de grundläggande elementen i FileMaker API för PHP, och grunderna i anpassad webbpublicering med PHP.

Beroende på vilka alternativ du anger kan du skapa sidor som gör att webbanvändarna kan göra följande:

- granska en post åt gången
- visa en lista över alla poster i databasen
- söka i databasen och visa resultatet i en lista
- sortera poster

- lägga till poster
- redigera och duplicera poster
- radera poster
- visa en statistikrapport

Du kan även skapa en startsida som är kopplad till andra webbplatsidor som har skapats. Mer detaljerad information finns i hjälpen för PHP Site Assistant.

Innan du använder PHP Site Assistant

Innan du använder PHP Site Assistant för att skapa din webbplats ska du göra följande:

1. Genomför alla steg som anges i kapitel 3, "Förbereda databaser för anpassad webbpublicering".
2. Med hjälp av FileMaker Server Admin Console kontrollerar du att webbservern och webbpubliceringsmotorn körs. Mer instruktioner finns i FileMaker Server Hjälp.
3. Aktivera anpassad webbpublicering med PHP i webbpubliceringsmotorn som du kommer att använda för att köra och testa PHP-koden. Mer instruktioner finns i FileMaker Server Hjälp.
4. När du ansluter till en databas från PHP Site Assistant använder du ett konto med en behörighetsuppsättning som stämmer överens med den du skulle tilldela webbplatsanvändarna. Se FileMaker Pro Hjälp för information om konton och behörighetsuppsättningar. Se PHP Site Assistant Hjälp för information om att ansluta till databasen.

Starta PHP Site Assistant

PHP Site Assistant är installerat på huvuddatorn i din FileMaker Server-driftsättning. Du kan använda PHP Site Assistant från vilken dator som helst med tillgång till huvuddatorn via nätverket. När du startar PHP Site Assistant använder FileMaker Server Java webbstartsteknik för att ladda ner PHP Site Assistant till din klientdator. PHP Site Assistant körs på din klientdator och ansluter endast till FileMaker Server när den behöver åtkomst till en databas.

Starta PHP Site Assistant

1. Öppna sidan med webbpubliceringsverktyg i FileMaker Server på något av dessa två sätt:
 - Starta en webbläsare och gå till <http://<server>:16000/tools> där <server> är huvuddatorn i din FileMaker Server-driftsättning.
 - I FileMaker Pro Advanced väljer du Arkiv-menyn > Fjärröppna fil för att öppna en databas som FileMaker Server är värd för och som har utökad fmphp-behörighet aktiverad. Välj Verktyg-menyn > Kör PHP Assistant.
2. På sidan med webbpubliceringsverktyg i FileMaker Server klickar du på Starta PHP Site Assistant. FileMaker Server laddar ner de efterfrågade filerna på din dator.
3. (Mac OS) Klicka på Trust för att fortsätta.
4. I dialogrutan skapa genväg klickar du på Ja för att skapa en genväg till PHP Site Assistant på datorn.
 - Windows: Genvägen sparas på skrivbordet.
 - Mac OS: Ange ett namn och en plats för genvägen och klicka på Spara.
5. Nu kan du använda PHP Site Assistant. Se PHP Site Assistant Hjälp för fullständiga instruktioner om hur du skapar en webbplats.

Obs!

- Om du vill starta PHP Site Assistant från FileMaker Pro Advanced måste du vara värd för den aktiva databasen på FileMaker Server och aktivera den utökade fmpHP-behörigheten i en behörighetsuppsättning för minst ett konto.
- Om IP-adressen för huvuddatorn i din FileMaker Server-driftsättning ändras kommer genvägarna inte längre att fungera. Följ instruktionerna ovan för att skapa nya genvägar.

Använda webbplatser som har skapats med PHP Site Assistant

Du kan använda PHP-koden som skapats av PHP Site Assistant utan att ändra den, eller så kan du använda dina egna PHP- eller textredigeringsverktyg för att lägga till flera funktioner och mer innehåll till den webbplats som har skapats. Du kan även utveckla din webbplats utan att använda PHP Site Assistant och införliva FileMaker API-funktioner i din PHP-kod för att få åtkomst till dina FileMaker-data. Se kapitel 5, ”Använda FileMaker API för PHP”.

Information om att starta och använda webbplatser som har skapats av PHP Site Assistant, se kapitel 6, ”Iscensätta, testa och övervaka en webbplats”.

Kapitel 5

Använda FileMaker API för PHP

I FileMaker API för PHP ingår en PHP-klass – FileMaker class – som ger ett objektorienterat gränssnitt för FileMaker-databaser. FileMaker API för PHP gör att man kan få åtkomst till både uppgifter och logik som har lagrats i FileMaker Pro-databaser och publicera dem på Internet, eller exportera dem till andra program. Med FileMaker API för PHP kan PHP-kod utföra samma sorts funktioner som du redan har tillgång till i FileMaker Pro-databaser:

- skapa, radera, redigera och duplicera poster
- utföra sökningar
- kontrollera fält och poster
- använda layouter
- köra FileMaker-manus
- visa portaler och relaterade poster
- använda värdelistor

I detta kapitel beskrivs hur man använder FileMaker class-objekt och -metoder för att lägga till dessa vanliga funktioner i en PHP-lösning. Detta kapitel täcker inte hela FileMaker API för PHP, men introducerar nyckelobjekt och -metoder.

Mer information

Mer information om FileMaker API för PHP finns på följande ställen.

Om du redan har en PHP-motor installerad och konfigurerad och du endast vill lägga till FileMaker API för PHP går du till ”Installera FileMaker API för PHP manuellt” på sidan 17.

Referens för FileMaker API för PHP

Om du installerade FileMaker API för PHP hittar du referensinformation i webbplatskomponenten i din FileMaker Server-driftsättning.

- För IIS (Windows):
<enhet>\Program\FileMaker\FileMaker Server\Documentation\PHP API Documentation\index.html
där <enhet> är den enhet där webbserverkomponenten för din FileMaker Server-driftsättning finns.
- För Apache (Mac OS): /Bibliotek/FileMaker Server/Documentation/PHP API Documentation/index.html

Introduktion till FileMaker API för PHP

Om du installerade FileMaker API för PHP hittar du en introduktion i webbplatskomponenten i din FileMaker Server-driftsättning.

- För IIS (Windows): <enhet>\Program\FileMaker\FileMaker Server\Examples\PHP\Tutorial
där <enhet> är den enhet där webbserverkomponenten för din FileMaker Server-driftsättning finns.
- För Apache (Mac OS): /Bibliotek/FileMaker Server/Examples/PHP/Tutorial

Om du vill vara värd för dessa PHP-introduktionsfiler kopierar du dem till webbserverns rotkatalog.

Exempel till FileMaker API för PHP

Om du installerade FileMaker API för PHP hittar du ytterligare exempel i webbplatskomponenten i din FileMaker Server-driftsättning.

- För IIS (Windows): <enhet>\Program\FileMaker\FileMaker Server\Examples\PHP\API Examples där <enhet> är den enhet där webbserverkomponenten i din FileMaker Server-driftsättning finns.
- För Apache (Mac OS): /Bibliotek/FileMaker Server/Examples/PHP/API Examples

Om du vill vara värd för dessa API-exempelfiler kopierar du dem till webbserverns rotkatalog.

Använda FileMaker klass

Om du vill använda FileMaker klass (class) i din PHP-lösning lägger du till följande uttryck i din PHP-kod:
`require_once ('FileMaker.php');`

FileMaker klass-objekt

FileMaker klass definierar klassobjekt som du kan använda för att hämta uppgifter från FileMaker Pro-databaser.

Klassobjekt	Använda objektet för att
FileMaker-databas	Definiera databasegenskaperna Ansluta till en FileMaker Pro-databas Skaffa information om FileMaker API för PHP
Kommando	Skapa kommandon som lägger till poster, raderar poster, duplicerar poster, redigerar poster, utför sökningar och utför manus.
Layout	Arbeta med databaslayouter
Post	Arbeta med postdata
Fält	Arbeta med fältdata
Relaterad uppsättning	Arbeta med portalposter
Resultat	Bearbeta posterna från en sökning
Fel	Kontrollera om ett fel har inträffat Behandla fel

FileMaker kommandoobjekt

FileMaker klass definierar ett baskommandoobjekt som du kan använda för att exemplifiera ett specifikt kommando och för att ange kommandots parametrar. För att utföra kommandot måste du använda metoden `execute()`.

FileMaker klass definierar följande specifika kommandon:

- Kommandot Lägg till
- Kommandot Sammanfattad sökning
- Kommandot Radera
- Kommandot Duplicera
- Kommandot Redigera

- Kommandot Utför sökning, kommandot Visa alla, kommandot Sök efter alla
- Kommandot Sökning, som läggs till i en sammanfattad sökning
- Kommandot Utför manus

Dessa kommandon beskrivs mer i detalj i följande avsnitt:

- ”Arbeta med poster” på sidan 31
- ”Köra FileMaker-manus” på sidan 33
- ”Utföra sökningar” på sidan 38

Ansluta till en FileMaker-databas

FileMaker klass definierar ett databasobjekt som du exemplifierar för att ansluta till en server eller en databas. Definiera objekttegenskaperna med klasskonstruktören, eller genom att använda metoden `setProperty()`.

Exempel: Ansluta till en server för att få en lista över databaser

```
$fm = new FileMaker();
$databases = $fm->listDatabases();
```

Exempel: Ansluta till en specifik databas på en server

Egenskaperna för användarnamn och lösenord avgör behörighetsuppsättningen för denna anslutning.

```
$fm = new FileMaker();
$fm->setProperty('database', 'questionnaire');
$fm->setProperty('hostspect', 'http://192.168.100.110');
$fm->setProperty('username', 'web');
$fm->setProperty('password', 'web');
```

Obs! Egenskapen `hostspect` har standardvärdet `http://localhost`. Om PHP-motorn körs på samma maskin som webbserverkomponenten i FileMaker Server-driftsättningen behöver man inte ange `hostspect`-egenskapen. Om PHP-motorn körs på en annan maskin använder man `hostspect` för att ange platsen för webbserverkomponenten i FileMaker Server-driftsättningen.

Arbeta med poster

FileMaker klass definierar ett postobjekt som du kan exemplifiera för att arbeta med poster. Ett exempel på ett postobjekt representerar en post från en FileMaker Pro-databas. Använd ett postobjekt med kommandona `Lägg till`, `Radera`, `Duplicera` och `Redigera` för att ändra uppgifterna i posten. Sökkommandona – `Utför sökning`, `Visa alla`, `Sök efter alla` och `Sammanfattad sökning` – returnerar en samling postobjekt.

Skapa en post

Det finns två sätt att skapa en post:

- Använd metoden `createRecord()` och ange ett layoutnamn, du kan även ange olika fältvärden. Du kan även ställa in värden individuellt i det nya postobjektet.

Metoden `createRecord()` sparar inte den nya posten i databasen. Om du vill spara posten i databasen använder du metoden `commit()`.

Till exempel:

```
$rec =& $fm->createRecord('Form View', $values);
$result = $rec->commit();
```

- Använd kommandot `Lägg till`. Använd metoden `newAddCommand()` för att skapa ett `FileMaker_Command_Add`-objekt, ange layoutnamnet och postdata. Om du vill spara posten i databasen använder du metoden `execute()`.

Till exempel:

```
$newAdd =& $fm->newAddCommand('Respondent', $respondent_data);
$result = $newAdd->execute();
```

Duplicera en post

Du kan duplicera en befintlig post med kommandot `Duplicera`. Använd metoden `newDuplicateCommand()` för att skapa ett `FileMaker_Command_Duplicate`-objekt, ange layoutnamn och post-ID för den post du vill duplicera. Duplicera sedan posten genom att använda metoden `execute()`.

Exempel

```
$newDuplicate =& $fm->newDuplicateCommand('Respondent', $rec_ID);
$result = $newDuplicate->execute();
```

Redigera en post

Det finns två sätt att redigera en post:

- Använda kommandot `Redigera`. Använd metoden `newEditCommand()` för att skapa ett `FileMaker_Command_Edit`-objekt, ange layoutnamn, post-ID för den post du vill redigera och värden som du vill uppdatera. Redigera sedan posten genom att använda metoden `execute()`.

Till exempel:

```
$newEdit =& $fm->newEditCommand('Respondent', $rec_ID, $respondent_data);
$result = $newEdit->execute();
```

- Använda ett postobjekt. Hämta en post från databasen, ändra fältvärdena och redigera sedan posten genom att använda metoden `commit()`.

Till exempel:

```
$rec = $fm->getRecordById('Form View', $rec_ID);
$rec->setField('Name', $nameEntered);
$result = $rec->commit();
```

Radera en post

Det finns två sätt att radera en post:

- Hämta en post från databasen och använd sedan metoden `delete()`.

Till exempel:

```
$rec = $fm->getRecordById('Form View', $rec_ID);
$rec->delete();
```

- Du kan radera en befintlig post med kommandot `Radera`. Använd metoden `newDeleteCommand()` för att skapa ett `FileMaker_Command_Delete`-objekt, ange layoutnamn och post-ID för den post du vill radera. Radera sedan posten genom att använda metoden `execute()`.

Till exempel:

```
$newDelete =& $fm->newDeleteCommand('Respondent', $rec_ID);
$result = $newDelete->execute();
```

Köra FileMaker-manus

Ett FileMaker-manus är en namngiven uppsättning ScriptMaker-manussteg. FileMaker class definierar olika metoder som gör att du kan arbeta med ScriptMaker-manus som har definierats i en FileMaker Pro-databas. Mer information om webbkompatibla manussteg (de manussteg som kan utföras i en webblösning) finns i "FileMaker-manus och anpassad webbpublicering" på sidan 22.

Hämta en lista på tillgängliga manus

Använd metoden `listScripts()` för att hämta en lista över tillgängliga manus från den aktuella anslutna databasen. Metoden `listScripts()` returnerar manus som kan utföras av det användarnamn och lösenord som angavs när databasanslutningen definierades. (Se "Ansluta till en FileMaker-databas" på sidan 31.)

Exempel

```
$scripts = $fm->listScripts();
```

Köra ett ScriptMaker-manus

Använd metoden `newPerformScriptCommand()` för att skapa ett `FileMaker_Command_PerformScript`-objekt, ange layout, manusnamn och alla manusparametrar. Utför sedan manuset genom att använda metoden `execute()`.

Exempel

```
$newPerformScript =& $fm->newPerformScriptCommand('Order Summary', 'ComputeTotal');
$result = $newPerformScript->execute();
```

Köra ett manus innan ett kommando utförs

Använd metoden `setPreCommandScript()` för att ange ett manus som körs innan ett kommando körs. Följande exempel använder ett sökkommando, men du kan använda metoden `setPreCommandScript()` med vilket kommando som helst.

Exempel

```
$findCommand =& $fm->newFindCommand('Students');
$findCommand->addFindCriterion('GPA', $searchValue);
$findCommand->setPreCommandScript('UpdateGPA');
$result = $findCommand->execute();
```

Köra ett manus innan en resultatuppsättning sorteras

Använd metoden `setPreSortScript()` för att ange ett manus som körs efter ett sökresultat har genererats, men innan resultatuppsättningen sorteras. Mer information finns i ”Använda kommandot Utför sökning” på sidan 39.

Exempel

```
$findCommand =& $fm->newFindCommand('Students');  
$findCommand->setPreSortScript('RemoveExpelled');
```

Köra ett manus efter resultatuppsättningen har genererats

Använd metoden `setScript()` för att ange ett manussteg som körs efter ett sökresultat har genererats. Mer information finns i ”Använda kommandot Utför sökning” på sidan 39.

Exempel

```
$findCommand =& $fm->newFindCommand('Students');  
$findCommand->setScript('myScript','param1|param2|param3');
```

Ordning för att utföra manus

Du kan använda metoderna `setPreCommandScript()`, `setPreSortScript()` och `setScript()` tillsammans med metoderna `setResultLayout()` och `addSortRule()` för ett enda kommando. FileMaker Server och webbpubliceringsmotorn bearbetar dessa metoder i följande ordning:

1. Kör det manus som har angetts för metoden `setPreCommandScript()` om detta har angetts.
2. Behandlar själva kommandot, exempelvis kommandona Utför sökning eller Radera post.
3. Kör det manus som har angetts för metoden `setPreSortScript()` om detta har angetts.
4. Sorterar sökresultatet, om metoden `addSortRule()` har angetts.
5. Gör att metoden `setResultLayout()` växlar till en annan layout, om detta har angetts.
6. Kör det manus som har angetts för metoden `setScript()` om detta har angetts.
7. Returnerar det slutgiltiga sökresultatet.

Om något av ovanstående steg genererar en felkod, stoppas kommandot och alla eventuella kvarvarande steg utförs inte. Eventuella tidigare steg i förfrågan utförs fortfarande.

Anta att ett kommando tar bort den aktuella posten, sorterar posterna och sedan kör ett manus. Om metoden `addSortRule()` anger ett fält som inte finns, tar förfrågan bort den aktuella posten och genererar felkod 102 (”Fältet saknas”), men manuset utförs inte.

Arbeta med FileMaker-layouter

En layout är en samordning av fält, objekt, bilder och layoutdelar som visar på vilket sätt information organiseras och presenteras när användaren bearbetar, granskar eller skriver ut poster. FileMaker klass definierar olika metoder som gör att du kan arbeta med de layouter som har definierats i en FileMaker Pro-databas. Du kan få information om layouter från flera av FileMaker klass-objekten.

Med detta klassobjekt Använd dessa metoder

Databas	<ul style="list-style-type: none"> listLayouts() hämtar en lista över tillgängliga layoutnamn. getLayout() hämtar ett layoutobjekt genom att ange ett layoutnamn.
Layout	<ul style="list-style-type: none"> getName() hämtar layoutnamnet för ett specifikt layoutobjekt. listFields() hämtar alla fältnamn som används i en layout. getFields() hämtar ett associativt urval av namn på alla fält som nycklar, och de associerade FileMaker_Field-objekten som matrisvärden. listValueLists() hämtar en lista över namn på värdelistor. listRelatedSets() hämtar en lista över namn på relaterade uppsättningar. getDatabase() returnerar namnet på databasen.
Post	<ul style="list-style-type: none"> getLayout() returnerar det layoutobjekt som är associerat med en specifik post.
Fält	<ul style="list-style-type: none"> getLayout() returnerar det layoutobjekt som innehåller ett specifikt fält.
Kommando	<ul style="list-style-type: none"> setResultLayout() returnerar kommandots resultat i en annan layout än den aktuella layouten.

Använda portaler

En portal är en tabell som visar rader med data från en eller flera relaterade poster. FileMaker klass definierar ett relaterat uppsättningsobjekt och flera metoder som gör att du kan arbeta med portaler som har definierats i en FileMaker Pro-databas.

Ett relaterat uppsättningsobjekt är en matris med postobjekt från den relaterade portalen – varje postobjekt representerar en rad med data i portalen.

Lista de portaler som har definierats på en specifik layout

För ett specifikt layoutobjekt använder du metoden listRelatedSets() för att hämta en lista över tabellnamn för alla portaler som har definierats i denna layout.

Exempel

```
$tableNames = $currentLayout->listRelatedSets();
```

Hämta portalnamn för ett specifikt resultatobjekt

För ett specifikt FileMaker_Result-objekt använder du metoden getRelatedSets() för att hämta namnen på alla portaler i denna post.

Exempel

```
$relatedSetsNames = $result->getRelatedSets();
```

Hämta information om portaler för en specifik layout

För ett specifikt layoutobjekt använder du metoden `getRelatedSets()` för att hämta `FileMaker_RelatedSet`-objekt som beskriver portalerna i layouten. Den returnerade matrisen är en associativ matris med tabellnamnen som matrisnycklar och de associerade `FileMaker_RelatedSet`-objekten som matrisvärden.

Exempel

```
$relatedSetsArray = $currentLayout->getRelatedSets();
```

Hämta information för en specifik portal

För ett specifikt layoutobjekt använder du metoden `getRelatedSet()` för att hämta det `FileMaker_RelatedSet`-objekt som beskriver den specifika portalen.

Exempel

```
$relatedSet = $currentLayout->getRelatedSet('customers');
```

Hämta tabellnamnet för en portal

För ett relaterat uppsättningsobjekt använder du metoden `getName()` för att få tabellnamnet för portalen.

Exempel

```
$tableName = $relatedSet->getName();
```

Hämta portalposterna för en specifik post

För ett specifikt postobjekt använder du metoden `getRelatedSet()` för att hämta relaterade poster för en specifik portal på den posten.

Exempel

```
$relatedRecordsArray = $currentRecord->getRelatedSet('customers');
```

Skapa en ny post i en portal

Använd metoden `newRelatedRecord()` för att skapa en ny post i den angivna relaterade uppsättningen och utför ändringen i databasen genom att använda metoden `commit()`.

Exempel

```
//skapa en ny portalrad i "kundportalen"
$new_row = $currentRecord->newRelatedRecord('customer');

//ställ in fältvärden på den nya portalraden
$new_row->setField('customer::name', $newName);
$new_row->setField('customer::company', $newCompany);

$result = $new_row->commit();
```

Ta bort en post från en portal

Använd metoden `delete()` för att ta bort en post i en portal.

Exempel

```

$relatedSet = $currentRecord->getRelatedSet('customers');
/* Kör igenom alla portalrader */
foreach ($relatedSet as $nextRow) {

    $nameField = $nextRow->getField('customer::name')
    if ($nameField == $badName ) {
        $result = $newRow->delete();
    }

}

```

Använda värdelistor

En värdelista är en uppsättning fördefinierade alternativ. FileMaker klass definierar olika metoder som gör att du kan arbeta med värdelistor som har definierats i en FileMaker Pro-databas.

Hämta namnen på alla värdelistor för en specifik layout

För en specifik layout använder du metoden `listValueLists()` för att hämta en matris som innehåller namnen på värdelistorna.

Exempel

```
$valueListNames = $currentLayout->listValueLists();
```

Hämta en matris över alla värdelistor för en specifik layout

För en specifik layout använder du metoden `getValueLists()` för att hämta en matris som innehåller värdena från alla värdelistorna. Den returnerade matrisen är en associerad matris. Matrisnycklarna är värdelistnamnen, och matrisvärdena är matriser som listar alternativen från varje värdelista.

Exempel

```
$valueListsArray = $currentLayout->getValueLists();
```

Hämta alternativen för en namngiven värdelista

För en specifik layout använder du metoden `getValueList()` för att hämta en matris som innehåller alternativen för den namngivna värdelistan.

Exempel

```
$valuesArray = $layout->getValueList('prefixes');
```

För ett specifikt fältobjekt använder du metoden `getValueList()` för att hämta en matris som innehåller alternativen för den namngivna värdelistan. - {} - Om fältet inte är ett värdelistefält returnerade denna metod ett nollvärde.

Exempel

```
$valuesArray = $currentField->getValueList('style');
```

Om databasen använder en värdelista som endast visar relaterade värden (värden ställs in dynamiskt baserade på värdena i ett annat fält), ska du ta med ett post-ID-värde i metoden `getValueList()` för att hämta värdelisteuppgifterna för en specifik post.

Exempel

```
$value = $layout->getValueList('style', $record->getRecordId());
```

Utföra sökningar

FileMaker klass definierar fyra olika sökkommandoobjekt:

- Kommandot Visa alla. Se ”Använda kommandot Visa alla” på sidan 39.
- Kommandot Sök efter alla. Se ”Använda kommandot Sök efter alla” på sidan 39.
- Kommandot Utför sökning. Se ”Använda kommandot Utför sökning” på sidan 39.
- Kommandot Sammanfattad sökning. Se ”Använda kommandot Sammanfattad sökning” på sidan 39.

FileMaker klass definierar även flera metoder som kan användas för alla fyra typer av sökkommandon:

- Använd metoden `addSortRule()` för att lägga till en regel som definierar hur sökresultatet ska sorteras. Använd metoden `clearSortRules()` för att rensa alla sorteringsregler som har definierats.
- Använd `setLogicalOperator()` för att växla mellan logiska AND-sökningar och logiska OR-sökningar.
- Använd metoden `setRange()` för att endast efterfråga en del av sökresultatet. Använd metoden `getRange()` för att hämta den aktuella intervalldefinitionen.

Genom att använda metoden `setRange()` kan man förbättra lösningens prestanda genom att minska antalet poster som returneras av sökförfrågan. Om en sökförfrågan exempelvis returnerar 100 poster kan du dela upp sökresultatet i fem grupper om 20 träffar istället för att behandla alla 100 poster på en gång.

- Du kan utföra ScriptMaker-manus tillsammans med sökkommandon.
 - Om du vill köra ett manus innan du utför sökkommandot använder du metoden `setPreCommandScript()`.
 - Om du vill köra ett manus innan du sorterar sökresultatet använder du metoden `setPreSortScript()`.
 - Om du vill köra ett manus efter ett sökresultat har genererats, men innan sökresultatet sorteras använder du metoden `setScript()`.

Använda kommandot *Visa alla*

Använd kommandot *Visa alla* för att hämta alla poster från en specifik layout. Använd metoden `newFindAllCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_FindAll`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

Exempel

```
$findCommand =& $fm->newFindAllCommand('Form View');
$result = $findCommand->execute;
```

Använda kommandot *Sök efter alla*

Använd kommandot *Sök efter alla* för att slumpmässigt hämta en post från en specifik layout. Använd metoden `newFindAnyCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_FindAny`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

Exempel

```
$findCommand =& $fm->newFindAnyCommand('Form View');
$result = $findCommand->execute;
```

Använda kommandot *Utför sökning*

Använd metoden `newFindCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_Find`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

Använd metoden `addFindCriterion()` för att lägga till villkor i sökförfrågan. Använd metoden `clearFindCriteria()` för att rensa alla sökvillkor som har definierats.

Exempel – Söka efter en post med fältnamn

```
$findCommand =& $fm->newFindCommand('Form View');
$findCommand->addFindCriterion('Questionnaire ID', $active_questionnaire_id);
$result = $findCommand->execute();
```

Exempel – Lägga till en sorteringsordning

```
$findCommand =& $fm->newFindCommand('Customer List');
$findCommand->addSortRule('Title', 1, FILEMAKER_SORT_ASCEND);
$result = $findCommand->execute();
```

Använda kommandot *Sammanfattad sökning*

Med kommandot *Sammanfattad sökning* kan du kombinera flera sökobjekt i ett enda kommando.

Så här skapar du kommandot *Sammanfattad sökning*:

- Skapa ett `FileMaker_Command_CompoundFind`-objekt genom att använda metoden `newCompoundFindCommand()`.
- Skapa ett eller flera `FileMaker_Command_FindRequest`-objekt genom att använda metoden `newFindRequest()`.
- Använd metoden `setOmit()` för att ange poster i sökresultatet för en specifik sökförfrågan som ska uteslutas från det slutgiltiga sökresultatet.
- Använd metoden `add()` för att lägga till sökobjekten i kommandoobjektet *Sammanfattad sökning*.
- Utför sedan den sammanfattade sökningen genom att använda metoden `execute()`.

Exempel – Kommandot Sammanfattad sökning

```
// Skapa kommandoobjektet Sammanfattad sökning
$compoundFind =& $fm->newCompoundFindCommand('Form View');

// Skapa den första sökförfrågan
$findreq1 =& $fm->newFindRequest('Form View');

// Skapa den andra sökförfrågan
$findreq2 =& $fm->newFindRequest('Form View');

// Skapa den tredje sökförfrågan
$findreq3 =& $fm->newFindRequest('Form View');

// Ange sökvillkor för den första sökförfrågan
$findreq1->addFindCriterion('Quantity in Stock', '<100');

// Ange sökvillkor för den andra sökförfrågan
$findreq2->addFindCriterion('Quantity in Stock', '0');
$findreq2->setOmit(true);

// Ange sökvillkor för den tredje sökförfrågan
$findreq3->addFindCriterion('Cover Photo Credit', 'The London Morning News');
$findreq3->setOmit(true);

// Ange sökförfrågningar till kommandot Sammanställd sökning
$compoundFind->add(1,$findreq1);
$compoundFind->add(2,$findreq2);
$compoundFind->add(3,$findreq3);

// Ange sorteringsordning
$compoundFind->addSortRule('Title', 1, FILEMAKER_SORT_DESCEND);

// Utför kommandot Sammanfattad sökning
$result = $compoundFind->execute();

// Hämta poster från sökresultatet
$records = $result->getRecords();

// Skriv ut antalet funna poster
echo 'Found ' . count($records) . " results.<br><br>";
```

Behandla posterna i ett sökresultat

- Hämta en matris med alla poster i sökresultatet genom att använda metoden `getRecords()`. Varje del i matrisen är ett `FileMaker_Record`-objekt, eller ett exempel på det klassnamn som har angetts i API för att exemplifiera poster. Matrisen kan vara tom om sökresultatet inte innehåller några poster.
- Hämta en lista över fältnamn för alla fält i sökresultatet genom att använda metoden `getFields()`. Metoden returnerar endast fältnamnen. Om du behöver ytterligare information om fälten använder du associerade layoutobjekt.
- Hämta antalet poster i hela sökresultatet genom att använda metoden `getFoundSetCount()`.
- Hämta antalet poster i det filtrerade sökresultatet genom att använda metoden `getFetchCount()`. Om inga parametrar har ställts in för sökkommandot är detta värde lika med resultatet med metoden `getFoundSetCount()`. Det är alltid lika med värdet för `count($response->getRecords())`.
- För en specifik post använder du metoden `getField()` för att returnera innehållet i ett fält som en sträng. Använd metoden `getFieldAsTimestamp()` för att returnera innehållet i ett fält som en Unix-tidsstämpel (PHP:s sätt att återge datum).
 - Om fältet är ett datumfält anges tidsstämpeln för fältdatumet vid midnatt.
 - Om fältet är ett tidsfält anges tidsstämpeln för den tiden den 1 januari 1970.
 - Om fältet är ett tidsstämpelfält kopplas FileMaker-tidsstämpelvärdet direkt till Unix-tidsstämpeln.
 - Om det angivna fältet inte är ett datum- eller tidsfält, eller om den genererade tidsstämpeln skulle ligga utanför intervallet, returnerar metoden `getFieldAsTimestamp()` ett `FileMaker_Error`-objekt.

Filtrera portalrader som returnerats av en sökförfrågan

I en lösning som har många relaterade poster kan sökning och sortering av portalposter vara tidskrävande. För att begränsa antalet poster som visas i ett sökresultat kan du använda metoden `setRelatedSetsFilters()` för sökresultat. Metoden `setRelatedSetsFilters()` använder två argument:

- en relaterad uppsättnings filtervärde: `layout` eller `inget`.
 - Om du anger värdet `inget` returnerar webbpubliceringsmotorn alla rader i portalen, och portalposterna sorteras inte i förväg.
 - Om du anger värdet `layout` respekteras inställningarna som har angetts i dialogrutan FileMaker Pro Tillval för portal. Posterna sorteras baserat på den sortering som har angetts i dialogrutan Tillval för portal, med postuppsättningen filtrerad för att starta med den första angivna raden.
- det maximala antalet portalposter som returneras: ett heltal eller `alla`.
 - Detta värde används endast om inställningen Visa vertikal rullningslist är aktiverad i dialogrutan Tillval för portal. Om du anger ett heltal returneras det antalet rader efter den första raden. Om du anger `alla` returnerar webbpubliceringsmotorn alla relaterade poster.
 - Om inställningen Visa vertikal rullningslist är inaktiverat kommer inställningen Antal rader i dialogrutan Tillval för portal att avgöra det maximala antalet relaterade poster som returneras.

Förkontrollera kommandon, poster och fält

Med FileMaker class kan du *förkontrollera* fältuppgifter i en PHP-lösning på webbservern innan du skickar uppgifterna till databasen.

När du avgör om du ska använda förkontroll eller inte, ska du tänka över antalet datavärden som webbanvändaren anger. Om användaren uppdaterar ett litet antal fält kan du förbättra prestandan genom att inte använda förkontroll. Om användaren däremot anger data i många fält kan förkontroll förhindra att användaren blir frustrerad över att en post avvisas av databasen på grund av kontrollfel.

Med FileMaker class förkontrollerar PHP-motorn följande fältvillkor:

- ej tomt
Giltiga data är en icke-tom teckensträng. Dessa data måste innehålla minst ett tecken.
- endast numeriskt
Giltiga data innehåller endast numeriska tecken.
- maximalt antal tecken
Giltiga data innehåller högst det antal tecken som angetts.
- fyrsiffrigt årtal
Giltiga data är en teckensträng som representerar ett datum med ett fyrsiffrigt årtal i formatet ÅÅÅÅ-D-M, där M är ett tal mellan 1 och 12, D är ett tal mellan 1 och 31 och ÅÅÅÅ är ett fyrsiffrigt tal mellan 0001 och 4000. Till exempel är 3030-30-1 ett giltigt fyrsiffrigt årtalsvärde. Däremot är, 2006-31-4 ett ogiltigt fyrsiffrigt årtalsvärde, eftersom april inte har 31 dagar. Datumkontroll stöder snedstreck (/), omvänt snedstreck (\) och bindestreck (-) som avgränsare. Strängen får dock inte innehålla en blandning av avgränsarna. Till exempel är 2008-30\1 ogiltigt.
- tidpunkt
Giltiga data är en teckensträng som representera ett 12-timmarsvärde i ett av dessa format:
 - T
 - T:M
 - T:M:S
 - T:M:S FM/EM
 - T:M FM/EM
 där T är ett tal mellan 1 och 12 och M och S är tal mellan 1 och 60.

Förkontrollen i PHP-motorn stöder implicit kontroll av fältdata baserat på fälttypen.

- datum
Ett fält om har definierats som ett datumfält kontrolleras enligt reglerna i ”fyrsiffrigt årtal”, förutom att årtalsvärdet kan innehålla 0-4 siffror (årtalsvärdet kan vara tomt). Till exempel är 30-1 ett giltigt datum även om det inte har något årtal angivet.
- tid
Ett fält som har definierats som ett tidsfält kontrolleras enligt reglerna för ”tidpunkt”, förutom att timkomponenten (T) kan vara ett tal mellan 1 och 24 för att stödja 24-timmarsvärden.
- tidsstämpel
Ett fält som har definierats som ett tidsstämpelfält kontrolleras enligt reglerna för ”tid” för tidskomponenten och enligt reglerna för ”datum” för datumkomponenten.

FileMaker class kan inte förkontrollera alla fältkontrollalternativ som är tillgängliga i FileMaker Pro. Följande kontrollalternativ kan inte förkontrolleras eftersom de är beroende av status för alla data i databasen vid det tillfälle då data verkställs:

- unikt värde
- befintligt värde
- i intervall
- medlem i värdelista
- kontrollera genom beräkning

Förkontrollera poster i ett kommando

För ett kommandoobjekt använder du metoden `validate()` för att kontrollera ett fält eller hela kommandot mot förkontrollsreglerna som drivs igenom av PHP-motorn. Om du godkänns av argumentet valfritt fältnamn förkontrolleras endast det fältet.

Om förkontrollen godkänns returnerar metoden `validate()` TRUE. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

Förkontrollera poster

För ett postobjekt använder du metoden `validate()` för att kontrollera ett fält eller alla fält i posten mot förkontrollsreglerna som drivs igenom av PHP-motorn. Om du godkänns av argumentet valfritt fältnamn förkontrolleras endast det fältet.

Om förkontrollen godkänns returnerar metoden `validate()` TRUE. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

Förkontrollera fält

För ett fältobjekt använder du metoden `validate()` för att avgöra om ett givet värde är giltigt för ett fält.

Om förkontrollen godkänns returnerar metoden `validate()` TRUE. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

Behandla kontrollfel

När förkontrollen inte godkänns innehåller det `FileMaker_Error_Validation`-objekt som returneras en matris med tre element för varje kontrollfel:

1. Det fältobjekt som inte kunde förkontrolleras

2. Ett konstant kontrollvärde som anger vilken kontrollregel som inte godkändes:

- 1 - FILEMAKER_RULE_NOTEMPTY
- 2 - FILEMAKER_RULE_NUMERICONLY
- 3 - FILEMAKER_RULE_MAXCHARACTERS
- 4 - FILEMAKER_RULE_FOURDIGITYEAR
- 5 - FILEMAKER_RULE_TIMEOFDAY
- 6 - FILEMAKER_RULE_TIMESTAMP_FIELD
- 7 - FILEMAKER_RULE_DATE_FIELD
- 8 - FILEMAKER_RULE_TIME_FIELD

3. Det verkliga värdet som angetts för det fält som inte kunde förkontrolleras

Du kan även använda följande metoder med FileMaker_Error_Validation-objekt:

- Använd metoden `isValidationError()` för att testa om felet är ett kontrollfel.
- Använd metoden `numErrors()` för att visa det antal kontrollregler som inte kunde godkännas.

Exempel

```
//Skapa en Lägg till-förfrågan
```

```
$addrequest =& $fm->newAddCommand('test', array('join' => 'added', 'maxchars' => 'abcx', 'field' => 'something',  
'numericonly' => 'abc'));
```

```
//Kontrollera alla fält
```

```
$result = $addrequest->validate();
```

```
//Om kontrolleringsmetoden() returnerade några fel, skriv ut namnet på fältet, felnumret och det värde som inte kunde  
godkännas.
```

```
if(FileMaker::isError($result)){  
    echo 'Validation failed:'. "\n";  
    $validationErrors= $result->getErrors();  
    foreach ($validationErrors as $error) {  
        $field = $error[0];  
        echo 'Field Name: ' . $field->getName(). "\n";  
        echo 'Error Code: ' . $error[1]. "\n";  
        echo 'Value: ' . $error[2]. "\n";  
    }  
}
```

Utenhet

Kontrollen misslyckades:

Fältnamn: numericonly

Felkod: 2

Värde: abc

Fältnamn: maxchars

Felkod: 3

Värde: abcx

Hantera fel

FileMaker class definierar FileMaker_Error-objekt för att hjälpa dig att hantera fel som uppstår i en PHP-lösning.

Ett fel kan uppstå när ett kommando körs. Om ett fel uppstår returnerar kommandot ett FileMaker_Error-objekt. Det är bra att kontrollera det fel som returnerar när ett kommando körs.

Använd följande metoder för att lära dig mer om felet som anges i FileMaker_Error-objektet.

- Testa om en variabel är ett FileMaker_Error-objekt genom att använda metoden `isError()`.
- Hämta antalet fel som uppstått genom att använda metoden `numErrors()`.
- Hämta en matris med matriser som beskriver de fel som uppstått genom att använda metoden `getErrors()`.
- Visa ett felmeddelande genom att använda metoden `getMessage()`.

Exempel

```
$result = $findCommand->execute();  
if (FileMaker::isError($result)) {  
    echo "<p>Error: " . $result->getMessage() . "</p>";  
    exit;  
}
```

Mer information om de felkoder som returneras med FileMaker_Error-objektet finns i bilaga A, ”Felkoder för anpassad webbpublicering med PHP”.

Kapitel 6

Iscensätta, testa och övervaka en webbplats

I detta kapitel finns instruktioner för att iscensätta och testa en webbplats med egen webbpublicering innan den körs i en produktionsmiljö. Instruktioner finns även för att använda loggfiler för att övervaka sidan vid testning och efter den har körts igång.

Iscensätta en anpassad webbpubliceringsplats

Innan du kan testa din webbplats ordentligt ska du kopiera och flytta de filer som behövs till korrekt plats på servern eller serverna där du ska iscensätta webbplatsen.

Så här iscensätter du och förbereder du webbplatsen för testning:

1. Genomför alla steg i kapitel 3, "Förbereda databaser för anpassad webbpublicering".
2. Kontrollera att webbservern och webbpubliceringsmotorn är i gång.
3. Kopiera eller flytta dina webbplatsfiler till webbserverkomponenten i din FileMaker-driftsättning.

Kopiera eller flytta dina webbplatsfiler till följande katalog på webbserverdatorn:

- IIS (Windows): <enhet>\inetpub\wwwroot där <enhet> är den enhet där webbpubliceringsmotorn till FileMaker Server-driftsättningen återfinns.
- Apache (Mac OS): /Bibliotek/WebServer/Documents

4. Om du inte redan har gjort det ska du kopiera eller flytta alla hänvisade containerobjekt till rätt katalog på webbserverdatorn.

Om ett containerfält i en databas lagrar en filreferens istället för en verklig fil kommer det hänvisade containerobjektet lagras i FileMaker Pro-katalogen när posten skapas eller redigeras. För att iscensätta din webbplats ska du kopiera eller flytta hänvisade containrar till en katalog med samma relativa position i rotkatalogen på webbserverprogramvaran.

Se "Publicera innehållet i containerfält på webben" på sidan 21.

5. Börja testet av webbplatsen.

Testa en anpassad webbpubliceringsplats

Innan du meddelar användarna att din anpassade webbpubliceringsplats är tillgänglig ska du kontrollera att den visas och fungerar som du har tänkt dig.

- Testa funktioner som sökning och att lägga till, radera och sortera poster med olika konton och behörighet.
- Kontrollera att behörighetsinställningarna fungerar som väntat genom att logga in med olika konton. Kontrollera att inga obehöriga användare kan komma åt och ändra dina data.
- Kontrollera alla manus så att du vet att resultatet blir det väntade. Information om hur du skapar manus som fungerar väl på webben finns i "FileMaker-manus och anpassad webbpublicering" på sidan 22.
- Testa webbplatsen med olika operativsystem och webbläsare.

Obs! Om du har installerat webbservern, webbpubliceringsmotorn och databasservern i en endatorslösning kan du visa och testa din webbplats utan att använda en nätverksanslutning. Flytta dina webbplatsfiler till rätt katalog på den datorn, och ange en av följande URL i din webbläsare:

- `http://localhost/<sökväg>`
- `http://127.0.0.1/<sökväg>`

Där `<sökväg>` är den relativa sökvägen till startsidan på din webbplats.

Övervaka webbplatsen

Du kan använda följande typer av loggfiler när du vill övervaka den egna webbpubliceringsplatsen och hämta information om de webbanvändare som besöker webbplatsen:

- Webbserveråtkomst och felloggar
- Programlogg för webbpubliceringsmotorn
- Fellogg för webbserververmodulen
- Webbpubliceringskärnans interna åtkomstloggar

Använda webbserveråtkomst och felloggar

- IIS (Windows): Webbservern Microsoft IIS genererar en åtkomstloggfil och visar felen i Loggboken (Windows) i stället för att skriva dem till en loggfil. Åtkomstloggfilen, som är i formatet W3C Extended Log File Format som standard, är en lista med alla inkommande HTTP-förfrågningar till webbservern. Du kan även använda formatet W3C Common Logfile Format för åtkomstloggen. Mer information finns i dokumentationen till Microsoft IIS-webbservern.
- Apache (endast Mac OS): Webbservern Apache genererar en åtkomstloggfil och en felloggfil. Åtkomstloggfilen på Apache, som är i formatet W3C Common Logfile Format som standard, är en post över alla inkommande HTTP-förfrågningar till webbservern. Felloggen på Apache är en lista med problem som rör bearbetningen av HTTP-förfrågningar. Mer information om dessa loggfiler finns i dokumentationen för Apache-webbservern.

Obs! Information om formaten W3C Common Logfile Format och W3C Extended Log File Format finns på webbplatsen www.w3.org.

Använda webbpubliceringsmotorns programlogg

Som standard genererar webbpubliceringsmotorn en programlogg som innehåller en post över fel, manus och användarens loggningsinformation. Loggfilen har namnet `pe_application_log.txt` och finns på webbpubliceringskomponenten i FileMaker Server-driftsättningen:

- IIS (Windows):
`<enhet>\Program\FileMaker\FileMaker Server\Logs\pe_application_log.txt`
där `<enhet>` är den primära enhet där programmet startas.
- Apache (Mac OS): `/Bibliotek/FileMaker Server/Logs/pe_application_log.txt`

Filen `pe_application_log.txt` skapas om något av följande loggningsalternativ är aktiverat i webbpubliceringsmotorn:

Loggningsalternativ aktiverat	Information registrerad i <code>pe_application_log.txt</code>
Felloggning	Alla ovanliga fel som har inträffat på webbpubliceringsmotorn. Vanliga felmeddelanden som rapporteras till webbanvändaren, t.ex. om att en viss databas inte är öppen, registreras inte i felloggfilen.
Manusloggning	Alla fel som genereras när webbanvändarna verkställer manus. Den kan exempelvis innehålla en lista över manussteg som ignoreras om de inte är webbkompatibla.

Som standard är båda alternativen aktiverade. Information om hur man ställer in dessa alternativ med hjälp av Admin Console finns i *FileMaker Server startbok*.

Obs! Posterna i programloggen tas inte bort automatiskt och med tiden kan filen därför bli mycket stor. Du kan spara utrymme på hårddisken genom att arkivera programloggfilen med jämna mellanrum.

Använda felloggen för webbservern

Om webbservern inte kan ansluta till webbpubliceringsmotorn skapar webbservermodulen en loggfil över eventuella fel som uppstår vid bearbetningen. Loggfilen har namnet `web_server_module_log.txt` och finns på webbpubliceringskomponenten i FileMaker Server-driftsättningen:

- IIS (Windows): `<enhet>\Program\FileMaker\FileMaker Server\Logs\web_server_module_log.txt` där `<enhet>` är den primära enhet där programmet startas.
- Apache (Mac OS): `/Bibliotek/FileMaker Server/Logs/web_server_module_log.txt`

Använda webbpubliceringskärnans interna åtkomstlogg

Som standardinställning genererar webbpubliceringskärnans komponent på webbpubliceringsmotorn en loggfil för att registrera åtkomst till webbpubliceringskärnan. Loggfilen har namnet `wpc_access_log.txt`. Den innehåller ett register över alla förfrågningar från slutanvändare att generera utdata för webbpublicering och använda FileMaker Server Direkt webbpublicering. Dessa förfrågningar dirigeras från webbservern direkt till webbpubliceringskärnan.

Eftersom FileMaker AP för PHP använder HTTP POST för att få åtkomst till webbpubliceringskärnan registrerar loggfilen inte uppgifter om PHP-förfrågningarna. Du kan använda loggfilen för att se när användarna har skickat sina förfrågningar.

Denna fil finns i följande katalog på webbpubliceringsmotorkomponenten i FileMaker Server-driftsättningen:

- IIS (Windows): `<enhet>\Program\FileMaker\FileMaker Server\Logs\wpc_access_log.txt` där `<enhet>` är den primära enhet där programmet startas.
- Apache (Mac OS): `/Bibliotek/FileMaker Server/Logs/wpc_access_log.txt`

Felsöka webbplatsen

Om du har problem med att visa eller använda webbplatsen ska du kontrollera följande:

- Den utökade behörighetsuppsättningen i databasen är inställd för anpassad webbpublicering med PHP och kopplad till ett användarkonto. Se ”Aktivera anpassad webbpublicering med PHP för databaser” på sidan 19.

- Databasen finns på FileMaker Server och är öppen. Se FileMaker Server Hjälp.
 - Kontonamnet och lösenordet till databasen är korrekta.
 - Webbservern och webbpubliceringsmotorn måste vara igång.
 - PHP-publicering är aktiverad i webbpubliceringsmotorn.
 - Öppna sidan FileMaker Server tekniktest i en webbläsare:
http://<server>:16000/test
- där <server> är den dator där FileMaker Server finns.
- Klicka på länken [Testa PHP anpassad webbpublicering](#) för att öppna en PHP-sida som har åtkomst till FMServer_Sample testdatabasen.

Mer information finns i *FileMaker Server startbok* och FileMaker Server Hjälp.

Bilaga A

Felkoder för anpassad webbpublicering med PHP

Webbpubliceringsmotorn stöder två typer av felkoder för egen webbpublicering:

- Databas- och dataförfrågningsfel. Webbpubliceringsmotorn genererar en felkod när data efterfrågas från en publicerad databas. FileMaker API för PHP returnerar denna felkod som ett FileMaker_Error-objekt. Se nästa avsnitt, "Felkodsnummer för FileMaker-databaser."
- PHP-fel. Dessa fel genereras och returneras av PHP-komponenter, inklusive cURL-modulen. Se "Felkodsnummer för PHP-komponenter" på sidan 57.

Felkodsnummer för FileMaker-databaser

Som utvecklare av en egen webbpubliceringslösning måste du kontrollera värdet för den returnerade felkoden och hantera det på rätt sätt. Webbpubliceringsmotorn hanterar inte databasfel.

Felnummer	Beskrivning
-1	Okänt fel
0	Inget fel
1	Användaren har avbrutit åtgärden
2	Minnesfel
3	Kommandot är inte tillgängligt (exempelvis fel operativsystem, fel läge osv)
4	Okänt kommando
5	Ogiltigt kommando (ett Tilldela fält-steg har exempelvis ingen beräkning angiven)
6	Filen är skrivskyddad
7	Minnet räcker inte.
8	Tomt resultat
9	Otillräcklig behörighet
10	Begärda data saknas
11	Namnet är ogiltigt
12	Namnet finns redan
13	Filen eller objektet används
14	Utanför det tillåtna intervallet
15	Går inte att dela med noll
16	Åtgärden misslyckades, begär nytt försök (t.ex. en användarfråga)
17	Försök att konvertera främmande teckenuppsättning till UTF-16 misslyckades
18	Klienten måste ange kontoinformation för att fortsätta
19	Strängen innehåller andra tecken än A-Z, a-z, 0-9 (Ascii)
100	Fil saknas

Felnummer	Beskrivning
101	Post saknas
102	Fält saknas
103	Relation saknas
104	Manus saknas
105	Layout saknas
106	Tabell saknas
107	Index saknas
108	Värdelista saknas
109	Behörighetsuppsättning saknas
110	Relaterade tabeller saknas
111	Fältrepetitionen är ogiltig
112	Fönstret saknas
113	Funktionen saknas
114	Filreferensen saknas
115	Menyuppsättningen saknas
116	Layoutobjektet saknas
117	Datakällan saknas
130	Filerna är skadade eller saknas och måste installeras om
131	Språkpaketfiler saknas (t.ex. mallfiler)
200	Tillgång till posten nekad
201	Det går inte att ändra fältet
202	Tillgång till fältet nekad
203	Det finns inga poster i filen att skriva ut eller så saknar lösenordet behörighet för utskrift
204	Ingen tillgång till fält i sorteringsordning
205	Användaren har inte behörighet att skapa nya poster; importen kommer att skriva över befintliga data
206	Användaren har inte behörighet att ändra lösenord eller så kan filen inte ändras
207	Användaren har inte tillräcklig behörighet för att ändra databasschemat eller så kan filen inte ändras
208	Lösenordet innehåller inte tillräckligt många tecken
209	Det nya lösenordet måste skilja sig från det befintliga
210	Användarkontot är inte aktivt
211	Lösenordet har upphört att gälla
212	Ogiltigt användarkonto och/eller lösenord. Försök igen
213	Användarkontot och/eller lösenordet finns inte
214	För många inloggningsförsök

Felnummer	Beskrivning
215	Administratörsbehörighet kan inte dupliceras
216	Gästkontot kan inte dupliceras
217	Användaren har inte tillräcklig behörighet för att ändra administratörskontot
300	Filen är låst eller används
301	Posten används av en annan användare
302	Tabellen används av en annan användare
303	Databasschemat används av en annan användare
304	Layouten används av en annan användare
306	Postens ändrings-ID matchar inte
400	Sökvillkoren är tomma
401	Inga poster matchar sökposten
402	Det valda fältet är inte ett matchande fält för en länk
403	Maxpostgränsen för testversionen av FileMakerPro har överskridits
404	Sorteringsordningen är ogiltig
405	Det angivna antalet poster överskrider antalet poster som kan utelämnas
406	Ogiltiga villkor för ersättning/ändring av löpnummer
407	Ett av eller båda de matchande fälten saknas (ogiltig relation)
408	Det angivna fältet har fel datatyp för den här åtgärden
409	Importordningen är ogiltig
410	Exportordningen är ogiltig
412	Fel version av FileMakerPro används för att reparera filen
413	Det angivna fältet är av fel typ
414	Resultatet kan inte visas i layouten
415	En eller flera nödvändiga relaterade poster är inte tillgängliga
416	En primär nyckel krävs från datakällstabellen
417	Databasen är inte en datakälla som stöds
500	Datumvärdet stämmer inte med kontrolltillvalen
501	Tidsvärdet stämmer inte med kontrolltillvalen
502	Det numeriska värdet stämmer inte med kontrolltillvalen
503	Värdet i fältet stämmer inte med intervallet som angetts i kontrolltillvalen
504	Värdet i fältet är inte unikt, vilket angetts som ett krav i kontrolltillvalen
505	Värdet i fältet är inte ett befintligt värde i databasfilen, vilket angetts som ett krav i kontrolltillvalen
506	Värdet i fältet finns inte med i värdelistan som angetts i kontrolltillvalet
507	Värdet i fältet klarade inte beräkningstestet för kontrolltillvalet

Felnummer	Beskrivning
508	Ett ogiltigt värde har angetts i sökläget
509	Fältet kräver ett giltigt värde
510	Det relaterade värdet är tomt eller inte tillgängligt
511	Värdet i fältet överskrider antalet tillåtna tecken
512	Posten har redan ändrats av en annan användare
513	För att skapa en post måste posten ha ett värde i minst ett fält
600	Utskriftsfel
601	Sidhuvudet och sidfoten ryms inte tillsammans på en sida
602	Huvuddelen ryms inte på en sida med den aktuella kolumninställningen
603	Ingen skrivaranslutning
700	Fel filformat för import
706	EPSF-filen saknar granskningsbild
707	Det går inte att hitta grafikkonverteraren
708	Det går inte att importera filen eller så krävs stöd för färgskärm för att importera filen
709	Import av QuickTime-film misslyckades
710	Det går inte att uppdatera QuickTime-filmreferensen eftersom databasfilen är skrivskyddad
711	Det går inte att hitta importkonverteraren
714	Lösenordet saknar behörighet för den här åtgärden
715	Det angivna Excel-kalkylbladet eller intervallet saknas
716	En SQL-fråga med DELETE, INSERT eller UPDATE är inte tillåten vid ODBC-import
717	Det finns inte tillräckligt med XML/XSL-information för att importen eller exporten ska kunna fortsätta
718	Ett fel inträffade vid tolkningen av XML-filen (från Xerces)
719	Ett fel inträffade vid konverteringen av XML med hjälp av XSL (från Xalan)
720	Ett fel inträffade vid exporten, det angivna formatet har inte stöd för repeterade fält
721	Ett okänt fel inträffade vid tolkningen eller konverteringen
722	Det går inte att importera data till en fil utan fält
723	Du har inte behörighet att lägga till eller ändra poster i måltabellen
724	Du har inte behörighet att lägga till poster i måltabellen
725	Du har inte behörighet att ändra poster i måltabellen
726	Importfilen innehåller fler poster än måltabellen. Alla poster importerades inte
727	Måltabellen innehåller fler poster än importfilen. Alla poster uppdaterades inte
729	Fel inträffade vid importen. Posterna kunde inte importeras
730	Excel-versionen stöds inte. Konvertera filen till Excel 7.0 (Excel 95), Excel 97, Excel 2000 eller Excel XP och försök igen

Felnummer	Beskrivning
731	Filen som du importerar från innehåller inga data
732	Den här filen kan inte sättas in eftersom den innehåller andra filer
733	En tabell kan inte importeras till sig själv
734	Den här filtypen kan inte visas som en bild
735	Den här filtypen kan inte visas som en bild. Den kommer att sättas in och visas som en fil.
736	Det finns för många data för att exportera till detta format. Data kommer att förkortas.
800	Det går inte att skapa filen på disken
801	Det går inte att skapa någon tillfällig fil på systemdisken
802	Det går inte att öppna filen. Detta fel kan bero på en eller flera av följande orsaker: <ul style="list-style-type: none"> ▪ Ogiltigt databasnamn ▪ Filen är stängd i FileMaker Server ▪ Ogiltig behörighet
803	Filen är en enanvändarfil eller så går det inte att hitta värden
804	Filen går inte att öppna som skrivskyddad i aktuellt tillstånd
805	Filen är skadad; använd kommandot Reparera
806	Det går inte att öppna filen i den här versionen av FileMaker Pro
807	Filen är inte en FileMakerPro-fil eller så är den allvarligt skadad
808	Det går inte att öppna filen eftersom behörigheten är skadad
809	Disken/volymen är full
810	Disken/volymen är låst
811	Det går inte att öppna den tillfälliga filen som en FileMaker Pro-fil
813	Postsynkroniseringsfel i nätverket
814	Det går inte att öppna filen eftersom maximala antalet tillåtna filer redan är öppna
815	Det går inte att öppna den länkade filen
816	Det går inte att konvertera filen
817	Det går inte att öppna filen eftersom den inte hör till denna lösning
819	Det går inte att spara en lokal kopia av en fjärrfil
820	Filen håller på att stängas
821	Frånkoppling framtvängd av värden
822	FMI-filer hittades inte; installera om saknade filer
823	Det går inte att göra filen till en enanvändarfil eftersom gäster är anslutna
824	Filen är skadad eller så är den inte en FileMaker-fil
900	Allmänt fel i stavningsmotorn
901	Standardordlistan är inte installerad
902	Det gick inte att starta hjälpsystemet

Felnummer	Beskrivning
903	Detta kommando kan inte användas i en delad fil
904	Kommandot kan bara användas i en fil som delas under FileMaker Server
905	Inget aktivt fält har valts och kommandot kan bara användas om det finns ett aktivt fält
906	Den aktuella filen är inte delad; kommandot kan endast användas för delade filer
920	Det går inte att initiera stavningsmotorn
921	Det går inte att öppna den egna ordlistan för redigering
922	Det går inte att hitta den egna ordlistan
923	Den egna ordlistan är skrivskyddad
951	Ett oväntat fel har inträffat
954	XML-syntax stöds inte
955	Inget databasnamn
956	Det maximala antalet databassessioner har överskridits
957	Motstridiga kommandon
958	Parameter saknas i fråga
1200	Allmänt beräkningsfel
1201	För få parametrar i funktionen
1202	För många parametrar i funktionen
1203	Oväntat slut på beräkningen
1204	Ett numeriskt värde, en textkonstant, ett fältnamn eller "" krävs
1205	Kommentaren avslutas inte med "*/"
1206	Textkonstanten måste avslutas med ett citationstecken
1207	Udda parentes
1208	Operator saknas, funktionen hittades inte eller "" förväntades inte
1209	Ett namn (t.ex. ett fältnamn eller ett layoutnamn) saknas
1210	Plugin-funktionen har redan registrerats
1211	Det är inte tillåtet att använda listor i den här funktionen
1212	En operator (t.ex. +, -, *) behövs här
1213	Den här variabeln har redan definierats i Let-funktionen
1214	AVERAGE, COUNT, EXTEND, GETREPETITION, MAX, MIN, NPV, STDEV, SUM och GETSUMMARY: ett uttryck påträffades där det bara behövs ett fält
1215	Den här parametern är en ogiltig Get-funktionsparameter
1216	Första argumentet i en GETSUMMARY måste vara ett statistikfält
1217	Brytfältet är ogiltigt
1218	Det går inte att utvärdera talet
1219	Ett fält kan inte användas i sin egen formel

Felnummer	Beskrivning
1220	Fälttypen måste vara normal eller beräkning
1221	Datatypen måste vara numerisk, datum, tid eller tidsstämpel
1222	Beräkningen får inte vara lagrad
1223	Funktionen som hänvisas till har inte implementerats än
1224	Den refererade funktionen finns inte
1225	Funktionen som hänvisas till stöds inte i denna kontext
1400	Det gick inte att initiera ODBC-klientdrivrutinen. Kontrollera att ODBC-klientdrivrutinerna är korrekt installerade.
1401	Det gick inte att ställa in miljön (ODBC)
1402	Det gick inte att frigöra miljön (ODBC)
1403	Det gick inte att koppla från (ODBC)
1404	Det gick inte att tilldela en anslutning (ODBC)
1405	Det gick inte att frigöra en anslutning (ODBC)
1406	Kontroll av SQL-API misslyckades (ODBC)
1407	Det gick inte att tilldela en instruktion (ODBC)
1408	Utökat fel (ODBC)
1409	Utökat fel (ODBC)
1410	Utökat fel (ODBC)
1411	Utökat fel (ODBC)
1412	Utökat fel (ODBC)
1413	Utökat fel (ODBC)

Felkodsnummer för PHP-komponenter

FileMaker API för PHP använder flera PHP-komponenter. Dessa PHP-komponenter kan returnera flera felkoder som inte har angetts i ovanstående lista.

Till exempel, om webbpubliceringskärnan eller FileMaker Server inte körs, kan du få cURL-modulfelet `CURLE_GOT_NOTHING` (52).

Mer information om PHP-relaterade felkoder finns på PHP-webbplatsen på <http://php.net>.

Register

A

- Admin Console 17, 19
- Anpassad webbpublicering
 - aktivera i webbpubliceringsmotorn 20
 - använda manus 22
 - begränsa åtkomst av IP-adresser på webbserver 20
 - definition 9
 - manus 24
 - med PHP 11
 - med XML 11
 - med XSLT 11
 - utökad behörighet för 19
 - översikt 9
- ansluta
 - till en FileMaker-databas 31
 - till en FileMaker-server 31
- använda
 - layouter 35
 - manus 33
 - portaler 35
 - poster 31
 - värdelistor 37
- användarnamn
 - ange för anpassad webbpublicering 19

B

- behandla ett sökresultat 41
- behörighet 21
- behörighetsuppsättning, tilldela för anpassad webbpublicering 19
- beständiga databassessioner 19, 21

C

- client URL library 16
- containerfält
 - hur webb användare använder data 22
 - publicera innehållet i 21
- cURL 16
- cURL-modulfel 57

D

- databas, skydda publicerad 20
- databasobjekt 31
- databassessioner, beständiga 19, 21

- datumfält 42
- datumrepresentation 41
- Dialogrutan Tillval för portal 41
- Direkt webbpublicering
 - definition 9
 - dokumentation 7
- dokumentation 7
- dokumentation online 7
- duplicera en post 32
- dynamisk IP-adress 16

E

- ej tomt fält 42
- elektronisk dokumentation 7
- endast numeriskt fält 42
- exempel till FileMaker Api för PHP 30
- extern SQL-datakälla 19

F

- fel
 - databas, felkodsnummer 51
 - hantera 45
 - loggfiler för webbserver 48
- felsöka webbplatser 49
- felsökning
 - Egna webbpubliceringsplatser 47
- FileMaker Api för PHP 11
 - definition 11
 - exempel 30
 - introduktion 29
 - manuell installation 17
 - referens 29
- FileMaker class 30
- FileMaker class-objekt 30
 - databas 31
 - post 31
 - relaterad uppsättning 35
- FileMaker kommandoobjekt
 - Duplicera 32
 - Kommandot Sammanfattad sökning 39
 - Kommandot Sök efter alla 39
 - Kommandot Utför sökning 38, 39
 - Kommandot Visa alla 39
 - Lägg till 32
 - Radera 33
 - Redigera 32

FileMaker Server

- dokumentation 7
- installera 7

FileMaker Server Admin

- se Admin Console

fält

- datum 42
- ej tomt 42
- endast numeriskt 42
- fysisiffrigt årtal 42
- maximalt antal tecken 42
- tid 42
- tidpunkt 42
- tidstämpel 42

fält med fysisiffrigt årtal 42

fält med maximalt antal tecken 42

fält med tidpunkt 42

fördelar med XML och XSLT 12

förkontroll 42

- datum 42
- ej tomt 42
- endast numeriskt 42
- fysisiffrigt årtal 42
- fält 43
- kommandon 42
- maximalt antal tecken 42
- poster 43
- tid 42
- tidpunkt 42
- tidstämpel 42

G

Gif-filer, publicera på webben 22

Gästkonto

- aktivering 21
- inaktivera 21

H

hantera fel 45

I

installation av FileMaker Api för PHP 17

installationsdokumentation 7

introduktion till FileMaker Api för PHP 29

iscensätta webbplatser 47

J

JDBC-dokumentation 7

Jpeg-filer, publicera på webben 22

K

Kommandot Duplicera 32

Kommandot Lägg till 32

Kommandot Radera 33

Kommandot Redigera 32

Kommandot Sök efter alla 39

Kommandot Utför sökning 39

Kommandot Visa alla 39

konton och behörighet

- aktivera för anpassad webbpublicering 19

Gästkonto 21

manus 23

kontroll 42

datum 42

ej tomt 42

endast numeriskt 42

fysisiffrigt årtal 42

fält 43

kommandon 42

maximalt antal tecken 42

poster 43

tid 42

tidpunkt 42

tidstämpel 42

kontroll av befintligt värde 43

kontroll av medlem i värdelista 43

kontroll av unikt värde 43

kontroll av värden i intervall 43

kontrollera genom beräkning 43

L

Latin-1-kodning 26

layouter 35

loggfiler 47, 49

beskrivning 48

web_server_module_log.txt 49

webbserveråtkomst 48

lösenord

- ange för egen webbpublicering 19

inget lösenord för inloggning 21

Ändra lösenord, manussteg 21

M

Mac OS X Server Admin 16

manuell installation av FileMaker Api för PHP 17

manus 33

- i egen webbpublicering 22

konton och behörighet 23

Logga in igen 21
 tips och råd 22
 Ändra lösenord 21
 Manuset Logga in igen 21
 metoden add() 39
 metoden addSortRule() 38
 metoden clearSortRules() 38
 metoden commit() 32
 metoden createRecord() 32
 metoden delete() 33, 36
 metoden getDatabase() 35
 metoden getErrors() 45
 metoden getFetchCount() 41
 metoden getField() 41
 metoden getFieldAsTimestamp() 41
 metoden getFields() 35, 41
 metoden getFoundSetCount() 41
 metoden getLayout() 35
 metoden getMessage() 45
 metoden getName() 35, 36
 metoden getRange() 38
 metoden getRecords() 41
 metoden getRelatedSet() 36
 metoden getRelatedSets() 35
 metoden getValueList() 37
 metoden getValueLists() 37
 metoden isError() 45
 metoden isValidationError() 44
 metoden listFields() 35
 metoden listLayouts() 35
 metoden listRelatedSets() 35
 metoden listScripts() 33
 metoden listValueLists() 35, 37
 metoden newAddCommand() 32
 metoden newCompoundFindCommand() 39
 metoden newDeleteCommand() 33
 metoden newDuplicateCommand() 32
 metoden newEditCommand() 32
 metoden newFindAllCommand() 39
 metoden newFindAnyCommand() 39
 metoden newFindCommand() 39
 metoden newFindRequest() 39
 metoden newPerformScriptCommand() 33
 metoden newRelatedRecord() 36
 metoden numErrors() 44, 45
 metoden setLogicalOperator() 38
 metoden setOmit() 39
 metoden setPreCommandScript() 33, 38
 metoden setPreSortScript() 34, 38
 metoden setProperty() 31
 metoden setRange() 38
 metoden setRelatedSetsFilters() 41
 metoden setResultsLayout() 35
 metoden setScript() 34, 38
 metoden validate() 43
 metoder
 add() 39
 addSortRule() 38
 clearSortRules() 38
 commit() 32
 CreateRecord() 32
 delete() 33, 36
 getDatabase() 35
 getErrors() 45
 getFetchCount() 41
 getField() 41
 getFieldAsTimestamp() 41
 getFields() 35, 41
 getFoundSetCount() 41
 getLayout() 35
 getMessage() 45
 getName() 35, 36
 getRange() 38
 getRecords() 41
 getRelatedSet() 36
 getRelatedSets() 35
 getValueList() 37
 getValueLists() 37
 isError() 45
 isValidationError() 44
 listFields() 35
 listLayouts() 35
 listRelatedSets() 35
 listScripts() 33
 listValueLists() 35, 37
 newAddCommand() 32
 newCompoundFindCommand() 39
 newDeleteCommand() 33
 newDuplicateCommand() 32
 newEditCommand() 32
 newFindAllCommand() 39
 newFindAnyCommand() 39
 newFindCommand() 39
 newFindRequest() 39
 newPerformScriptCommand() 33
 newRelatedRecord() 36
 numErrors() 44, 45
 setLogicalOperator() 38

- setOmit() 39
- setPreCommandScript() 33, 38
- setPreSortScript() 34, 38
- setProperty() 31
- setRange() 38
- setRelatedSetsFilters() 41
- setResultsLayout() 35
- setScript() 34, 38
- validate() 43

N

- nummer för
 - databaser, felkoder 51

O

- ODBC-begränsningar 19
- ODBC-dokumentation 7

P

- PHP
 - fel 57
 - felsökning 50
 - fördelar 12
 - sammanfattning av steg för publicering 25
 - versioner som stöds 17
 - webbplatstestning 47
- PHP 5 16
- PHP Api för egen webbpublicering 11
- PHP Site Assistant
 - använda kod som skapats 28
 - förbereda för användning 27
 - starta 27
- portaler 35
- poster 31
- postobjekt 31
- programlogg 48
- publicera på webben
 - använda PHP 25
 - containerfältobjekt 21
 - databaser, felkoder 51
 - QuickTime-filmer 22
 - skydda databaser 20

Q

- QuickTime-filmer, publicera på webben 22

R

- radera en post 33

- redigera en post 32
- referensinformation 29
- relaterat uppsättningsobjekt 35

S

- Sammanfattad sökning
 - exempel 40
 - kommando 39
- SAT
 - se Admin Console
- serverkrav 16
- Site Assistant *Se* PHP Site Assistant
- Site Assistant. *Se* PHP Site Assistant
- skapa en post 32
- skapa PHP-kod 28
- skydda publicerade databaser 20
- SSL-kryptering 20
- statisk IP-adress 16
- statisk publicering, definition 9
- säkerhet
 - begränsa åtkomst från IP-adresser 20
 - dokumentation 11
 - konton och lösenord 20
 - riktlinjer för att skydda publicerade databaser 20
- Sökkommandoobjekt 38
- sökresultat 41

T

- tekniktest 50
- testa PHP-publicering 50
- testa webbplatser 47
- tidsfält 42
- tidsstämpelfält 41, 42

U

- Unicode 26
- Unix-tidsstämpel 41
- UTF-8-kodning 26
- utföra sökningar 38
- utökad behörighet för egen webbpublicering 19

V

- Verktyget Server Admin.
 - Se* Mac OS X Server Admin
- värdelistor 37

W

- Web Publishing Engine
 - felkoder 51
 - programlogg 48
- web_server_module_log.txt, loggfil 49
- webbanvändare
 - använda containerfältdata 22
- webbläsare
 - ta emot utdata 10
- webbmapp, kopiera containerfältobjekt 21
- webbplatser
 - felsökning 49
 - FileMakers supportsidor 7
 - iscensätta 47
 - testa 47
 - övervaka 48
- Webbpubliceringskärna
 - interna åtkomstloggar 49
- Webbpubliceringsmotor
 - beskrivning 10
- webbserver
 - loggfiler 48
- wpc_access_log.txt, fil 49

X

- XML egen webbpublicering 11
- XSLT egen webbpublicering 11

Å

- åtkomstloggfiler för webbserver, beskrivning 48

Ä

- Ändra lösenord, manussteg 21

Ö

- översikt
 - Anpassad webbpublicering 9
 - PHP-publicering 25
- övervaka webbplatser 48

