

Using PHP Site Assistant to create sites for mobile devices

Overview

In the last few years, a major shift has occurred in the number and capabilities of mobile devices. Improvements in processor speed, display quality and wireless networking make these devices useful for a wider range of tasks than ever before. Armed with new tools and technologies, developers are creating web-based applications for these new devices that blur the lines between desktops and handhelds.

For example, FileMaker Server 9 and FileMaker Server 9 Advanced provide the ability to interact with FileMaker files over the web via the new FileMaker API for PHP. FileMaker now provides a new application, PHP Site Assistant, to quickly create FileMaker-driven websites. The PHP Site Assistant presents a series of questions about how users will interact with the database over the web, and then builds a site that can be used as-is or customized. This paper addresses the modification of the pages generated by PHP Site Assistant to provide the best experience on mobile devices like Smartphones.

Important: This document is not intended to replace the product documentation. See the FileMaker Server documentation for complete instructions.

Differences between mobile and desktop devices

Each year the gap between desktop and mobile devices continues to narrow. Mobile processor speed and network connectivity continue to improve, making web browsing on these devices a viable alternative to laptops. Previously, mobile web browsers required a special markup language, which added significant development time and effort to deploying web applications for mobile devices. Fortunately, mobile web browsers now use standard HTML. Given all the advancements, the biggest limitation is the form factor itself—the size of the screen and input mechanism for data.

Although most mobile devices use a keyboard to enter information, the size reduces the usability and efficiency for entering large amounts of data. One of the ways developers can help their end users work more efficiently and accurately is by providing pop-up menus and radio buttons to reduce the amount of typing. You'll find more FileMaker design tips in the section "Preparing a database for mobile web access" below.

Another limitation inherent in mobile devices is the small screen size. This must be taken into consideration during database and site design. When designing an

application for a desktop computer, developers can present screens with a standard resolution, minimally 800x600 pixels. Unfortunately, there are no standards when designing a display for mobile devices. The screens on modern mobile devices range from 240x240 to 320x480. They may have objects like onscreen keyboards or button bars taking up room as well. It's important to design an interface that wraps well and displays legibly on a range of screen sizes, as described in "Modifying the Site Assistant's output to target mobile devices."

Popular mobile devices

The scope of this section is limited to the following mobile device families:

- Apple iPhone
- Windows Mobile devices
- Palm OS devices

Apple iPhone

The iPhone is an Internet-enabled quad-band GSM EDGE-supported mobile phone with 802.11g wireless networking. User input comes through a multi-touch screen with a virtual keyboard and buttons. The iPhone's display dimensions are 320x480 and while web browsing, the display can be rotated 90 degrees to allow pages to be viewed in widescreen mode. A zoom in/out feature also allows the user to see more data on the screen at once. The iPhone uses Safari as its web browser.

Windows Mobile

Windows Mobile is a handheld operating system that runs on a wide variety of mobile devices. Wireless networking, user input and screen resolution will vary by device so consult the manufacturer's documentation for specific information. The current version of the OS is Windows Mobile 6, which comes with a version of Internet Explorer designed specifically for Windows Mobile. Other browsers can also be loaded.

Palm OS

Palm is both a hardware and software provider and markets handheld devices that run Palm OS and Smartphones that run either Palm OS or Windows Mobile. As with the Windows Mobile devices screen resolution, networking and user input options vary from device to device, but specific device information can be found at the Palm website. Palm OS comes pre-loaded with its own web browser named Blazer, but other browsers can be installed.



Preparing a database for mobile web access

Before using PHP Site Assistant, there are some things that should be done within the database. Taking these steps now saves time and improves site performance.

- Designate layouts that will only be accessed by web-based users, and not FileMaker Pro users. When FileMaker data is requested from a web browser, the layout must also be included in the request. FileMaker sends data from every field on that specified layout to the web publishing engine, then on to the PHP application on the web server. It is important that the layout web users are interacting with only contains fields that are critical to the mobile users. This eliminates unnecessary overhead on the web server.
- Before running PHP Site Assistant, format the fields on your designated web layouts to display the same way they will on a web page. For example, first format a Gender field to display radio buttons with the "Male" and "Female" choices on the FileMaker layout. When PHP Site Assistant generates the site, it automatically creates the appropriate HTML to display the field the same way. Use pop-up menus and radio buttons wherever appropriate to make data entry easier for users.
- Because the FileMaker API for PHP also targets a specific FileMaker layout when searching a database, it is important to restrict the number of users who can modify these layouts. If a field is unknowingly removed from a layout, the web-based search could produce unexpected results.
- Create separate accounts and privilege sets for web-based access. This privilege set should only provide access to those layouts designated for web-based users. Similarly, the Extended Privileges should only allow access via PHP web publishing. Restricting ways that web-based users can access your databases reduces your security risks.
- Whether users are accessing data using FileMaker Pro or a web browser, performance will always be better if the fields you're working with are indexed. Wherever possible make sure that sorts and finds are not being performed on fields where indexing is turned off or that use or reference unstored calculations.

Getting started with PHP Site Assistant

The PHP Site Assistant is included in FileMaker Server 9 and FileMaker Server 9 Advanced. It uses a wizard-like interface that presents a series of questions about the type of site to create, the layouts to base the site on, and the way the site should look. It then generates a series of PHP pages that will perform the requested actions. The resulting pages can be customized with a web page editor or even a simple text editor like Notepad. Do not use a word processor, however, as it will not save the file back correctly. Customizing the PHP pages to improve the user experience on mobile devices is critical.

Before opening PHP Site Assistant, ensure the following:

1. The database must be hosted on FileMaker Server 9 or FileMaker Server 9 Advanced.
2. Custom Web Publishing with PHP must be enabled.
3. The database must contain at least one account belonging to a privilege set that has the **fmphp** extended privilege enabled.

Once the server and database are properly configured, PHP Site Assistant can be opened in two different ways:

- In a web browser, access the Web Publishing Tools start page:
`http://<server host name or IP address>:16000/tools`
Once the next page loads click the Start PHP Site Assistant button.
- Use FileMaker Pro 9 Advanced to open a file from the FileMaker 9 Server then go to the Tools menu > Launch PHP Site Assistant. This will open a web browser to the PHP and XSLT Site Assistant Tools page. Click the Start **PHP Site Assistant** button. If the button is not enabled, make sure you have the **fmphp** Extended Privilege enabled on a user account.

Both approaches will download PHP Site Assistant to the local machine. Subsequently, PHP Site Assistant can be opened directly from the local file.

Creating a site for mobile devices

After launching PHP Site Assistant, users create a new site and provide information about the capabilities their site will have, for example whether a site will be searched. The different screens in PHP Site Assistant are listed below along with important considerations when designing a site for use on mobile devices.



- Database Settings: The web server address, log in information, database and security.
- Layout Group: A set of layouts that all use the same table occurrence.
- Site Profile: Specifies the functions people need to perform. After selecting a Site Profile, the pages that make up that particular profile are displayed, along with options for setting the layout, fields and associated options for each page. Some best practices:
 - Limit the number of records being returned on the Record List page. Setting a low number of rows to display per page minimizes scrolling and improves load time.
 - Keep page titles as short as possible. The title of each page appears in the navigation bar at the top of the screen. Short titles reduce the chances the navigation bar will wrap onto multiple lines.
- Theme: Determines the look of the site. Choose Plain - White for the best experience on handheld computers.
- Output Settings: Allows you to preview the site and select where to save the files.

After PHP Site Assistant generates the site, the files are saved in the location specified on the Output Settings screen.

A look at the files PHP Site Assistant creates

When PHP Site Assistant generates the site, it creates several .php files and one .css file that reflects the theme that was specified. For example if the Plain - White theme was chosen a file named "plain_white.css" is generated.

If a theme with a background image was chosen, you'll also see an "Images" directory that contains the background images.

Each .php file contains a line in the header referencing the name of the .css file. For example, with the Plain - White theme, you'll see the following in the .php file:

```
<link rel="stylesheet" type="text/css" media="screen" href="plain_white.css">
```

Changing the theme in Site Assistant to Circles - Green and generating the site again changes the line in each .php file to:

```
<link rel="stylesheet" type="text/css" media="screen" href="circles_green.css">
```



Modifying the output of the Site Assistant to target mobile devices

Sites created by PHP Site Assistant are designed for desktop browsers, but with a few simple changes, the same site can serve a dual purpose, supporting both desktop and mobile browsers. This is accomplished by creating alternate style sheets designed specifically for mobile devices and by making minor changes to the .php pages.

Creating an alternate style sheet

The easiest way to create a style sheet specifically for mobile devices is to modify an existing style sheet already created by PHP Site Assistant. While any of the themes offered by PHP Site Assistant can be modified for this use, the Plain - White theme is recommended because it creates a site with a very basic design.

After PHP Site Assistant generates the pages in the site, open the site folder. Duplicate the plain_white.css file and rename the duplicate mobile.css. Once the general changes have been made to the mobile.css style sheet, you can duplicate it and use it as a starting point for the iPhone style sheet.

Choosing a theme in PHP Site Assistant does not change the functionality of the PHP code that is generated. The only exception is a single line that each .php file contains in its <head> section showing the style sheet it references. If the Plain - White theme was chosen, each of the .php files would contain this line:

```
<link rel="stylesheet" type="text/css" media="screen" href="plain_white.css">
```

Setting mobile devices to use the correct style sheet

The media ="screen" portion of the line above tells the browser to use the plain_white.css style sheet to display this page if it is being viewed on a desktop computer. Adding a second line containing media="handheld" tells the browser to use a different style sheet if the page is being viewed on a device that identifies itself as a handheld device. Assuming that a style sheet named "mobile.css" has already been created, adding this line to the header will direct handheld browsers to use the "mobile.css" style sheet:

```
<link rel="stylesheet" type="text/css" media="handheld" href="mobile.css">
```

Targeting the iPhone with a specific style sheet

This method is sufficient for most phones and PDAs, including Windows Mobile and



Palm devices, but not the iPhone. On the iPhone, the Safari web browser identifies itself as a desktop browser so it uses the style sheet specified for the "screen" media type.

In the iPhone developer documentation, Apple provides a technique for making the iPhone use a different style sheet by evaluating the maximum screen size of the device. Adding the following third line to each .php file causes the iPhone to use the mobile.css style sheet.

```
<link media="only screen and (max-device-width: 480px)" href="mobile.css" type="text/css" rel="stylesheet">
```

iPhone viewport

In addition to adding the above line, one more option must be set to ensure that the page displays at the correct scale. On the iPhone, the viewport determines the size at which a web page should be displayed. By default, the iPhone assumes that web pages are designed for a desktop browser and sets the viewport to 980 pixels wide.

Add the following line to the <head> section of each .php page to set the viewport to be 320 pixels wide:

```
<meta name = "viewport" content = "width = 320">
```

For additional information about setting viewport options see "Configuring the Viewport" in the Safari web content guide available from the Apple Developer Connection.

<http://developer.apple.com/iphone/devcenter/>

Changing the mobile.css style sheet

When making changes to the .css and .php pages, be aware that some mobile browsers do not support all features of CSS. Be sure to test your pages on the target mobile devices.

This section provides general guidelines on changing the style sheet generated by PHP Site Assistant. It is important that you do independent research to learn more about working with cascading style sheets.

All examples shown below are taken from code generated by PHP Site Assistant 1.0.

Padding

The Padding attribute determines the amount of space on each side of an element within that element's borders. Because mobile devices have much smaller screens than desktop devices, the padding should be set to the minimum value that ensures legibility.

Padding around many elements such as the header, the page navigation buttons, and the field data should be reduced to save screen space. In general, the padding on the top and bottom can safely be set to 0, but 1 or two pixels on the left and right may be necessary to avoid running the text together.

Margins

The Margin attribute is very similar to the Padding attribute except that the margin defines the white space around an element's border, while padding refers to the white space within the border.

The margin attributes in the style sheets modified for mobile devices can generally be set 0, and any white space needed can be provided using the padding attribute.

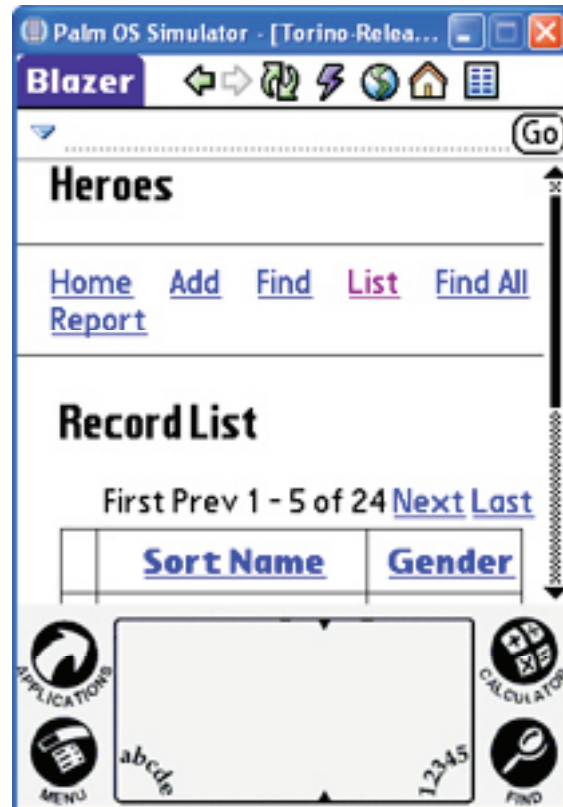
Hiding sections of the page

While some parts of the page should be modified to fit on the screen, others should be excluded on mobile devices. This can be accomplished by setting the display property to none. For example, the code shown below hides #header, which would normally appear above the page navigation buttons.

```
#header{
    display: none;
}
```

The images below show the difference these changes make. Both images show a list of five records, but the one on the right is rendered using a style sheet optimized for mobile devices





Changes to the individual .php pages

findrecords.php

If the site profile specified in PHP Site Assistant contains a search page, you can modify the findrecords.php page to reduce the number of records returned. At the bottom of the findrecords.php is a menu that controls the number of records displayed per page. The smallest value in the menu by default is 10. Because of the limited amount of space on mobile device screens it is advisable to add another choice to that menu so users can choose to return only 5 records.

recordlist.php

Move the "Restore Original Sort Order" link in the recordlist.php file out of the header and below the table to save space.

Working with images

There are three ways to serve images to the Web from FileMaker. The pros and cons of each must be evaluated to determine the best choice for a given application.

1. Use a container field to store the images inside the FileMaker database.

Pros

- Easy for FileMaker users to add and remove images.
- FileMaker and web users will see the images immediately.

Cons

- Web users will be unable to add or remove images.
- Database file size will increase quickly.

2. Use a container field and store a reference to the image file in FileMaker. The images must be stored on your web server.

Pros

- FileMaker users will see the images immediately.
- Database file size will not increase significantly.

Cons

- Web users will be unable to add and remove images.
- FileMaker users will need to have access to the web server volume
- If the files are moved or the path to them is changed, FileMaker and web users will lose access to them.
- More difficult approach than storing images in a container field.

3. Use a text field to store a URL to where the image resides.

Pros

- Database file size will not increase significantly.

Cons

- FileMaker users will need a scripted process to add and remove images.
- FileMaker users must use the Web Viewer to see the images.
- If the files are moved or the path to them is changed, FileMaker and web users will lose access to them.
- The files PHP Site Assistant generates will need to be modified in order to display the image instead of the text of the URL.

No matter how you store the images, you should optimize them for handheld devices.

- They should be in the GIF or JPEG format
- They should be relatively low resolution
- The image dimensions should fit on the screen without resizing



While these conditions do not have to be met in order for the images to be viewed, adhering to them will improve performance.

Refer to these documents for more information on working with images. The documents are included with FileMaker Server or can be downloaded from: www.filemaker.com/support/product/documentation.html

- FileMaker Server 9 Custom Web Publishing with PHP
- FileMaker API for PHP Tutorial

Testing your site

There is no single program that can be used to ensure that content displays correctly on all devices, but the manufacturers of each of the devices discussed provide test environments. These programs are free, but may require registration as part of a developer program in order to download them.

iPhone

Because the iPhone uses the same web browser that runs on OS X, Safari can be used on a desktop computer for testing. Apple is developing a version of Safari for Windows that can be downloaded, as a beta, from the Apple website.

Although running Safari on the iPhone is very similar to running it on the desktop there are a few differences. Safari on iPhone does not support the following web technologies:

- Modal dialogs
- Mouse-over events
- Hover styles
- Tooltips
- Java applets
- Flash
- SVG
- XSLT
- Plug-in installation
- Custom x.509 certificates
- WML
- File uploads and downloads

Palm OS

Palm Inc. offers the Palm Simulator to members of its developer program, which can be joined at <https://pluggedin.palm.com/regac/pluggedin/>. Simulators are offered for each of its Smartphone models that run the Palm OS.

Palm distributes Smartphones that run on either the Palm OS or Microsoft Windows Mobile. Downloads for both platforms are available from the Palm site. If the file has "Simulator" in the name, it is for a Palm OS device and should contain everything needed to run it. If the download is referred to as an "Emulator", it is for a Windows Mobile device and only contains a ROM file. To test on the Windows Mobile devices you will also need the Mobile Device Emulator from Microsoft. See the next section for more information.

Windows Mobile

Microsoft provides an application for testing software developed for Windows Mobile called Mobile Device Emulator. It is available in the Windows Mobile area of Microsoft's website. Be sure to get the standalone edition and to download both the efp.msi and the V1Emulator.zip files.

In addition to the Mobile Device Emulator, Microsoft Virtual PC 2007 will also need to be downloaded and installed. The Virtual PC application does not need to be running in order to run Mobile Device Emulator, but it must be installed in order to allow Mobile Device Emulator to use your machine's network connection.

As of this writing all of the device images that come with Mobile Device Emulator are for Windows Mobile 5. The current version of Windows Mobile is version 6. Therefore it is recommended that you also search for and download the Windows Mobile 6 localized emulator images from Microsoft's site.

About the author

Alan Kirtlink is an information analyst with Skeleton Key, a FileMaker Business Alliance member based in St. Louis, Missouri (<http://www.skeletonkey.com>). He has been developing in FileMaker and showing others how to do it since 1994. Alan is a FileMaker certified developer and trainer as well as a contributor to FileMaker Advisor magazine.



Additional resources

- FileMaker - PHP Overview
<http://www.filemaker.com/support/technologies/php.html>
- FileMaker - Documentation
<http://www.filemaker.com/support/product/documentation.html>
- Apple Developer Connection - Development for iPhone
<http://developer.apple.com/iphone/devcenter/>
- Microsoft Windows Mobile
<http://www.microsoft.com/windowsmobile/developers/>
- Palm Developer Network
<https://pdn.palm.com/regac/pdn/>
- W3C Mobile Best Practices
<http://www.w3.org/TR/mobile-bp/>

