Developer Handbook

Creating Applications Using M-Business Anywhere

Version 5.5

iAnywhere SOLUTIONS™ A SYBASE COMPANY

Company information

About iAnywhere®

iAnywhere Solutions, a subsidiary of Sybase, Inc. (NYSE: SY), provides software and services that enable the delivery of enterprise information anywhere, at anytime. More than 12,000 customers and 1,000 partners worldwide rely on the company's industry leading SQL Anywhere® database and mobile enterprise solutions. In addition, the company's AvantGo® mobile Internet service has more than ten million registered subscribers. Visit http://www.ianywhere.com or http://www.sybase.com for more information.

Contacting iAnywhere, Inc.

If you need assistance using iAnywhere software, please contact iAnywhere at 800-8SYBASE (800-879-2273); services will be provided in accordance with your support agreement.

If you need help with customizing iAnywhere software to better serve your enterprise, please contact iAnywhere Solutions Professional Services at contact_us@ianywhere.com.

If you need information about other iAnywhere products for your enterprise, please contact iAnywhere Workforce Sales at contact_us@ianywhere.com.

If you have questions or suggestions about this document or other iAnywhere technical publications, please contact iAnywhere Technical Publications at iasdoc@ianywhere.com.

Developer Handbook: Creating Applications Using M-Business Anywhere, Version 5.5

A portion of this software is based in part on the work of the Independent JPEG Group and on the work of the Netscape Communications' Netscape Communicator & Directory Server 5.0 (Mozilla) open source project.

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org).

Written and designed at iAnywhere, Inc.
One Sybase Drive
Dublin, CA 94568
http://www.ianywhere.com

550-046B0404D

Table of Contents

CHAPTER 1. About this handbook

Focus of this handbook

This handbook provides guidelines on developing mobile applications with M-Business Anywhere and M-Business Anywhere client. This handbook provides the information needed to understand the feature set that is available, the best approaches to using these features to develop mobile applications, and where to find detailed instructions for the various tasks involved.

This handbook is intended as a high-level overview of creating mobile applications using M-Business Anywhere and M-Business Anywhere client. Its objective is to quickly orient you to the development environment that M-Business Anywhere offers. Once oriented, these guidelines direct you to other M-Business Anywhere documentation for detailed implementation steps.

Conventions

The following table lists the formatting conventions used throughout this handbook.

Table 1-1 Formatting conventions

Item	Treatment	Example
Name of publication	Italic	*Administrator Guide for M-Business Anywhere*
User interface items: buttons, links, keywords	Bold	Click the **Reset** button.
Multi-level menu selections	Bold with vertical hash	Select **Start|Settings|Control Panel**.
Text you type	Courier	Type `Admin` in this field.
Text displayed in a file or on the screen	Courier	The screen reads: `Backup Complete`
Keyboard command key	Angle brackets	<Enter>
File names	Italic	*pods.h* *AGAppAlertServer.exe*
Code	Courier	Use `createStdArray()` to create an array to pass to JavaScript.
Variables in code	Courier italic	*`size`* in the code below: `PODSArray createStdArray( PODSUInt32 size);`
Attribute and method parentage	Double colon	`PODSObjectMgr::objectForName()`
Method names (vs. attribute names) in text and headings	Parentheses following name	Use `appendSubmissionElement()` to add the submission element to the submission.
Literals in code	Straight quotes	`CreateObject( "Acme.Socket")` ...a POD providing objects of class `"Acme.Socket".`

Related publications

In addition to this document, there are several other iAnywhere, Inc. publications available that you may find useful in developing mobile applications in the M-Business Anywhere environment.

Note

Unless otherwise noted, all of these publications are available from: http://www.ianywhere.com/developer/product_manuals/mbusiness_anywhere/

- *M-Business Anywhere Release Notes*
- *Administrator Guide for M-Business Anywhere*
- *User Guide for M-Business Client*
- *Channel Developer Guide for M-Business Anywhere*
- *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*
- *Ensuring Mobile Security from the Device to the Datacenter*, available from http://www.ianywhere.com/whitepapers/ensuring_security.html

Recommended references

In addition to the related publications from iAnywhere, Inc., you may want to consult references on the standards and third party software that are incorporated in the M-Business Anywhere architecture.

Note

The *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API* documents in detail only those features of JavaScript, DOM, and CSS that differ from the published standards.

If you already have a favorite reference on any of these topics, there is no need to seek out another. However, if you need additional reference material and do not already have it at hand, the following list of sources is a good place to start.

Cascading Style Sheets (CSS)

W3C Cascading Style Sheets home page.
HTML: http://www.w3.org/Style/CSS/
A comprehensive reference on CSS maintained by the W3C.

Document Object Model (DOM)

W3C Document Object Model (DOM) Level 1 Specification (Second Edition)
PDF: http://www.w3.org/TR/2000/WD-DOM-Level-1-20000929/DOM.pdf
plain text: http://www.w3.org/TR/2000/WD-DOM-Level-1-20000929/DOM.txt
A basic reference on the DOM spec.

DOM Tutorial
HTML: http://www.w3schools.com/dom/default.asp
A good introduction to DOM maintained by W3Schools.

Dynamic HTML (DHTML)

Introduction to Dynamic HTML .
HTML: http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml.asp
The Microsoft Developer Network's tutorial/reference on DHTML.

HTML

Dynamic HTML, the Definitive Reference
Author: Danny Goodman
Publisher: O'Reilly, August 1998
Amazon.com online description: http://www.amazon.com/exec/obidos/tg/feature/-/6779/103-8587514-0449445
A good source of examples, with tutorials, organized more for learning than for reference; content is targeted at desktop browsers.

JavaScript

Netscape Client-Side JavaScript Reference
HTML: http://devedge.netscape.com/library/manuals/2000/javascript/1.3/reference/
A basic reference on JavaScript; free online, but not completely up-to-date.

JavaScript Bible, 4th Edition
Authors: Danny Goodman and Brendan Eich
Publisher: Hungry Minds, Inc., April 2001
Publisher's online description: http://catalog.hungryminds.com/product.asp?isbn=0764533428
A more current popular basic reference on JavaScript, containing numerous coding examples.

XHTML

XHTML™ 1.0: The Extensible HyperText Markup Language
HTML: http://www.w3.org/TR/xhtml1/
A comprehensive reference on XHTML maintained by the W3C.

XHTML Tutorial
HTML: http://www.w3schools.com/xhtml/default.asp
A good introduction to XHTML maintained by W3Schools.

CHAPTER 2. M-Business Anywhere

Overview

The basic characteristics of wired and wireless environments dictate that applications for each need to be fundamentally different in design: batch or real-time processing. But the business world currently is operating in both environments, steadily moving to wireless as the cost comes down. To further complicate this division between wired and wireless, users of wireless devices may occasionally need to operate offline, either when service is interrupted or just to save connect charges.

M-Business Anywhere allows developers to write applications that run properly in both wired and wireless environments. M-Business Anywhere also allows end users to move freely between wired and wireless devices, and wireless workers can continue working in an application when disconnected. M-Business Anywhere provides an environment that:

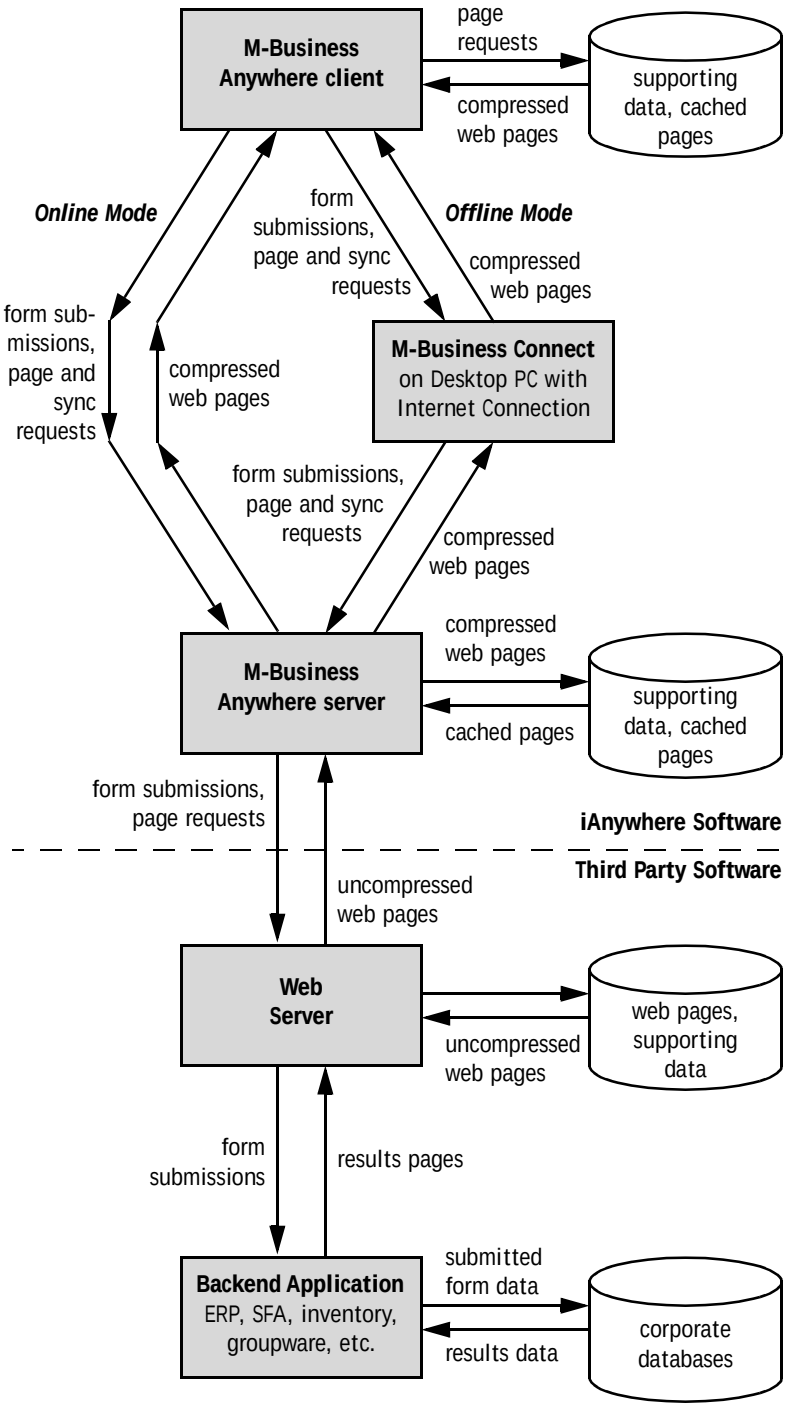
- Supports both wired and wireless users.
- Makes it possible to write applications that both groups can use.
- Allows wired users to work in offline mode whenever they want or need to do so.
- Permits users to easily migrate between online and offline modes without a learning curve.
- Supports a variety of platforms in both the wired and wireless environments.

For more information on creating applications using M-Business Anywhere, see “Custom branding for M-Business Anywhere client” (page 27) and “Applications can work both online and offline” (page 19).

Architecture

A diagram illustrating the M-Business Anywhere architecture is presented in Figure 2-1. The text that follows describes the key components.

Figure 2-1
Architecture of M-Business Anywhere



In the descriptions below, the bold headings correspond to the key components of M-Business Anywhere, which also appear in bold in the diagram in Figure 2-1 (page 13).

M-Business Anywhere client

M-Business Anywhere client is the user interface component of M-Business Anywhere. M-Business Anywhere client runs on a user's mobile device, either wired or wireless. Its primary function is to display the web pages that are downloaded through M-Business Anywhere. To do this, M-Business Anywhere client stores supporting data about the user and the M-Business Anywhere server(s) to which the user regularly connects.

M-Business Anywhere client also can store or cache downloaded pages on the device, so that a page that has not changed can be displayed from the cache instead of being downloaded from M-Business Server again. This reduces the load on your M-Business Server. For more information on caching downloaded pages on-device, see "Client-side caching" (page 22).

M-Business Anywhere client operates in one of two different modes: offline or online. A wired device can operate only in offline mode, but a wireless device can operate in either online or offline mode. Table 2-1 summarizes the characteristics of these modes. For more details on these modes, see "Offline mode" (page 15) and "Online mode" (page 16).

For general user instructions for M-Business Anywhere client, see the *User Guide for M-Business Client*.

Table 2-1 Summary of offline and online mode characteristics

Function	Offline mode	Online mode
Syncing files	Periodically, user must dock the mobile device with an Internet-connected PC and sync files on the mobile device with those on a server.	No docking is necessary; user is connected all the time and most files stay in sync. User syncs only to update subscribed channels, or after working in offline mode.
Completing transactions	Transaction data piles up while user is not connected, then is submitted in a batch during a sync. Results pages are returned in the same sync.	Transactions are real-time; transaction data is sent to the server as soon as user submits it and results pages are returned immediately.

Function	Offline mode	Online mode
Surfing web pages	User can surf only those web pages that have been downloaded to the device in a sync. Other pages requested must be downloaded in next sync.	User can surf anywhere on the Internet.
Pushing content	Server can only push content to a user only during a sync.	Server can push content to a user as soon as it is available.

Offline mode

In offline mode, also referred to as disconnected mode, the user operates a mobile device that does not have an active Internet connection. This may be a wired device, which the user periodically has to dock with an Internet-connected PC, in order to sync files on the mobile device with those on a server. Or it may be a wireless device that is being operated without its Internet connection.

In offline mode, M-Business Anywhere client queues up all forms that the user submits, plus any requests for pages that are not available on the device. M-Business Anywhere client sends queued form submissions and page requests to an M-Business Anywhere server when the user syncs the device. During a sync, in addition to processing submitted forms and downloading specific pages that the user has requested, M-Business Server downloads any updated pages in the channels that the user is subscribed to. For an explanation of what subscribed channel web pages are downloaded during a sync, see “Setting link depth” (page 20).

A sync is strictly a batch process; the user cannot interactively surf web pages or submit forms while connected to the server in a sync process. The user only can surf web pages that have been downloaded to the device in a sync, and M-Business Server only can push content to the user during a sync. Transaction data piles up while the user is not connected, then is submitted in a batch during a sync.

Without a wireless connection, the user initiates a sync from M-Business Anywhere client while the device is docked with an Internet-connected desktop PC that has M-Business Connect software installed. M-Business Connect handles all communication between an M-Business Anywhere client and an M-Business Anywhere server. M-Business Connect relays the sync request from an M-Business Anywhere client to the designated M-Business Anywhere server, and then relays the downloaded web pages (or error messages encountered) from the M-Business Anywhere server to the M-Business Anywhere client.

During a sync for a user operating in offline mode, three things happen:

- Any forms that have been submitted offline are sent to the host web server and results pages are returned from the web server to an M-Business Anywhere client.
- Any pages that the user has attempted to click through to that were not available on the offline device are also downloaded.
- Any updated pages in channels to which the user is subscribed are downloaded to the device.

After a sync is completed, the offline user may view and interact with any of the pages that were downloaded.

Note The user may select from several M-Business Servers that are available, and an application may switch servers transparently to the user. For more information on operating in offline mode, see the *User Guide for M-Business Client*. For information on programming an application to switch servers, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*.

Online mode

In online mode, also referred to as connected mode, the user operates a mobile device that has a wireless Internet connection and that connection is active. The user is continuously connected, except when service is interrupted or the user intentionally disconnects to save connection charges — then the M-Business Anywhere client automatically switches the user to offline mode.

In online mode, the M-Business Anywhere client submits forms and page requests to the connected M-Business Server in real time, as the user interacts with web pages. M-Business Server also returns in real time the pages that result from form submissions, plus the pages requested by the user.

If an online mode user works offline for a while, for example, when service is interrupted, the M-Business Anywhere client automatically switches to offline mode and queues up any form submissions and page requests. The user then initiates a sync when the wireless connection is re-established and the queued forms and page requests are submitted to M-Business Server. In order to update pages from subscribed channels, the online mode user must always initiate a sync with the associated M-Business Server.

Note The user may specify that the M-Business Anywhere client automatically should establish a connection when the device is offline and a user action or application process requires a connection. For user instructions on setting up the auto-connect option, see the *User Guide for M-Business Client*. For information on programming an application connect automatically, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*.

M-Business Connect

M-Business Connect runs on a PC with an Internet connection. The sole function of M-Business Connect is to link an M-Business Anywhere server with an M-Business Anywhere client on mobile devices that have no wireless connection. When the mobile device is docked with an Internet-connected PC, M-Business Connect relays the sync request from an M-Business Anywhere client to the associated M-Business Anywhere server, then relays downloaded pages from the M-Business Anywhere server to the M-Business Anywhere client. For more information on this process, see “Offline mode” (page 15).

M-Business Anywhere server

The M-Business Anywhere server is at the center of the iAnywhere architecture. Although only one M-Business Anywhere server is shown in Figure 2-1 (page 13), multiple M-Business Anywhere servers are supported and users can switch among them through M-Business Anywhere client menus; or, applications can switch among them transparently. For user instructions on switching servers, see the *User Guide for M-Business Client*. For information on programming an application to switch servers, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*.

The M-Business Anywhere server receives page and sync requests from M-Business Anywhere client users. These requests are received through M-Business Connect on an Internet-connected desktop PC, where either a wired user has docked a device, or the requests are received directly from a wireless user working in online mode. The M-Business Anywhere server only downloads pages which are not already present on the requesting device, or which have been updated since the copy of the page on the device was downloaded. After the M-Business Anywhere server determines which pages need to be downloaded, it gathers them from the associated web servers (or from its own page cache), and sends them on to the requesting device.

To speed up processing, the M-Business Anywhere server compresses all web pages that it receives. The web pages that the M-Business Anywhere server sends to users are highly compressed (much smaller size) copies. The M-Business Anywhere client decompresses these pages before they are displayed. To further speed up processing, and to reduce the load on the associated web servers, the M-Business Anywhere server can store

these compressed copies in a cache and send the stored copies to users, instead of downloading the pages again from the host web server. For more information, see “Server-side caching” (page 22).

The supporting data that the M-Business Anywhere server stores is information on users, channels, and user subscriptions to channels. The channels you define on an M-Business Anywhere server determine what content is available for users to subscribe to. For a description of how channels work, see “Understanding channels” (page 20).

For more information on how interaction with an M-Business Anywhere server works for wired users as compared with wireless users, see “Offline mode” (page 15) and “Online mode” (page 16).

Web servers

Web servers host the web pages that an M-Business Anywhere server sends to an M-Business Anywhere client. You can have as many web servers as needed within your company, in addition to making use of any that are accessible on the Internet. Web servers can be located anywhere, within or outside of your firewall.

Note If a web server is located within a firewall, users must also be set up within that firewall in order to access the server.

The content provided by a web server may be static HTML pages, or dynamic pages that are generated from a backend application. The type of content that M-Business Anywhere client can display is limited more by the constraints of the mobile device than by M-Business Anywhere client: small screens and limited color depth, or no color at all. M-Business Anywhere client can provide an interface to just about any website’s pages, if the those pages are designed with mobile devices in mind.

For information on tools available for use in web pages for the M-Business Anywhere client, see “Tools for use in HTML pages” (page 24). For guidance on designing HTML pages for the M-Business Anywhere client, see the *M-Business Channel Developer Guide*, referenced in “Related publications” (page 8).

Note You can greatly reduce the load on web servers that an M-Business Anywhere server accesses by implementing caching. See “Server-side caching” (page 22).

Backend applications

Many backend applications, such as enterprise resource planning (ERP) originated in the days of corporate mainframes. With the advent of powerful PCs, local area networks, and web browsers, today most backend applications can be run on a LAN server and have some sort of web interface. The M-Business Anywhere client can provide an interface to just about any backend application for which a web interface is available, if the web pages in that interface are designed with mobile devices in mind.

For information on tools available for use in web pages for the M-Business Anywhere client, see “Tools for use in HTML pages” (page 24). For guidance on designing HTML pages for the M-Business Anywhere client, see the *M-Business Channel Developer Guide*, referenced in “Related publications” (page 8).

Applications can work both online and offline

A key feature of the M-Business Anywhere architecture is that mobile applications can work in either online or offline modes. Wireless users can switch between modes as the need arises.

If an M-Business Anywhere client application is designed for operation in offline mode, it automatically will work properly when used in online mode. You can program your application to behave differently in online and offline modes if you like, but no such special programming is required.

For more guidance on designing applications to work both online and offline, see, “Designing applications to work both online and offline” (page 53).

Understanding channels

Channels, set up on M-Business Server, determine what web content an M-Business Anywhere client user can subscribe to. The channel definition on an M-Business Anywhere server specifies the following key parameters for the channel:

- **Top-level URL** - the URL for a single, top level web page that will be downloaded to the mobile device when subscribed users sync
- **Content settings** - a variety of settings that control the download of content to M-Business Anywhere clients. For example, setting that imposes a total size limit on the material that can be downloaded from the channel or whether binary downloads are permitted
- **Link depth** - the number of links away from the top-level page that M-Business Server should follow, downloading the additional web pages along with the channel's top-level web page

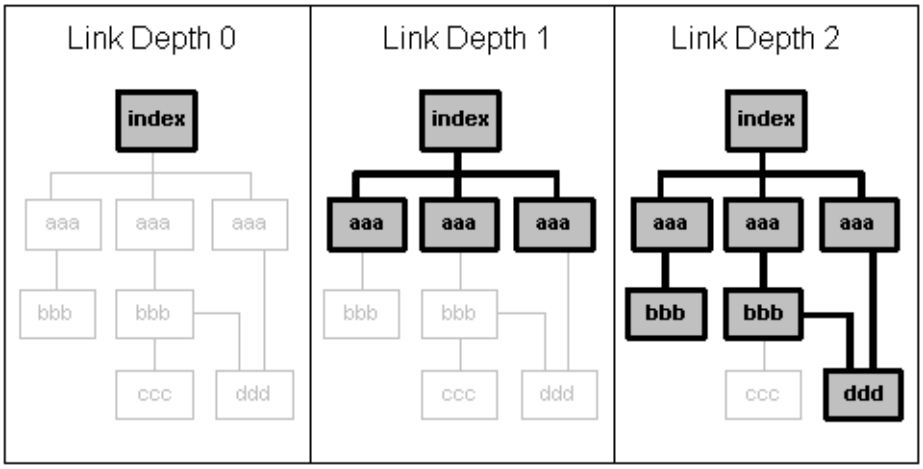
Setting link depth

Link depth is critical for users in offline mode because it determines how far from the top level page they can browse. Link depth is the number of links away from the channel's top-level web page that an M-Business Anywhere server downloads in a sync.

With link depth set to 0, only the top-level web page is downloaded. Setting link depth to 1 causes any pages linked directly from the top-level web page to be downloaded as well. A link depth of 2 adds pages that are linked directly from those pages linked from the top-level web page.

Figure 2-2 shows a typical HTML link hierarchy, indicating which pages would be downloaded with different link level settings. The page labeled "index" is the top-level web page specified in the channel definition.

Figure 2-2 Link depth illustrated graphically



In an enterprise setting, most channels are defined on an M-Business Anywhere server by the system administrator and are available to users that are also defined on the same server. If the M-Business Anywhere server allows it, users also can define their own private channels, which are accessible only by the defining user. For information on setting up channels, see “Setting up a channel and subscribing users” (page 57).

Subscribing users

After channels are set up, users must be subscribed to them before they can view channel content. The system administrator for an M-Business Anywhere server can subscribe users to channels, or the system administrator can set up individual channels so that users can subscribe themselves, either through the M-Business Anywhere server’s web interface or directly from an M-Business Anywhere client through a menu option. For information on setting up channels, see “Setting up a channel and subscribing users” (page 57).

Caching web pages for better performance

Caching is the process of storing a local copy of remote information, such as web pages, to save time accessing that information the next time it is needed. The M-Business Anywhere architecture supports caching of web pages on both the M-Business Anywhere server and on the M-Business Anywhere client.

Server-side caching

If you anticipate heavy volumes of traffic on your channel pages, the M-Business Anywhere server supports caching that can reduce the number of channel-related accesses to the supporting web server by up to 98%, in comparison with the same access volume handled without caching. Even if a web server is operated by a third party, a shared cache, as the server-side cache is called, still provides your company benefits by reducing the wait time that your end users experience when loading cached pages.

When you implement a shared cache, the M-Business Anywhere server acts as a proxy server for the web server(s) hosting your channels. When a cached page is requested, the M-Business Anywhere server follows rules that you have specified. The M-Business Anywhere server serves the cached page directly, if the page is not expired. If the page is expired, the M-Business Anywhere server checks with the web server to determine whether a new copy needs to be downloaded. If the web server indicates that a new copy of the page needs to be downloaded, the M-Business Anywhere server updates its cached copy from the web server and sends the page on to the user. If the web server indicates that that the cached copy of the page is still good, the M-Business Anywhere server updates the cached copy's expiration date accordingly and sends the cached copy on to the user.

For guidelines on setting up server-side caching in the M-Business Anywhere server, see "Implementing server-side caching" (page 32).

Client-side caching

The M-Business Anywhere client also has the ability to cache pages on the mobile device, which can further reduce the load on the web server(s) for your channels, reduce the load on your M-Business Anywhere server, and greatly reduce average load times for cached pages on the device. In addition, if the mobile device is operating in offline mode, client caching also makes the cached pages available while the user is offline.

On the server side, you can dedicate a huge amount of disk space to the shared cache. On the client side, local storage is at a premium. The effectiveness of the client cache is limited by the amount of memory that can be dedicated to caching.

The user controls the maximum size of the client cache. The user can also clear the client cache to make room in memory for a very large

page download, either directly in online mode, or during a sync in offline mode. All other parameters configuring the client cache are set by the system administrator for the M-Business Anywhere server.

For guidance on M-Business Anywhere server system administrator configuration of the client cache of users, see “Implementing client-side caching” (page 34). For more information on user control of the client cache, see the *User Guide for M-Business Client*, referenced in “Related publications” (page 8).

Tools for use in HTML pages

The M-Business Anywhere client supports a rich palate of features that web designers can use to create effective HTML pages. To make more effective use of the limited memory available on mobile devices, some of the least used features found in desktop browsers such as Microsoft's Internet Explorer have been omitted. But the vast majority of features that are used in the vast majority of websites are fully supported.

Web designers for mobile devices will find that they can continue to use most of the features they are accustomed to using in pages designed for desktop browsers. The greatest challenge is posed by inherent limitations of mobile devices: small screens, limited color depth, or no color at all.

There is also a far greater variation in the way a page displays on different mobile devices, as compared with the variations in how the same page may display on PCs running different browsers on different monitors. For guidance on designing HTML pages for the M-Business Anywhere client, see the *M-Business Channel Developer Guide*, referenced in "Related publications" (page 8).

Dynamic HTML (DHTML)

The combination of HTML 4 with JavaScript provides Dynamic HTML (DHTML). The HTML 4 specification includes Cascading Style Sheets (CSS) 1.0, Document Object Model (DOM) Level 1, and XHTML 1.0. The M-Business Anywhere client supports a rich subset of DHTML, giving developers the ability to create desktop-style applications in a browser window on a mobile device.

DHTML allows the state to be maintained across pages. DHTML lets page elements change dynamically with user interaction, without the need to download additional material from the server. And DHTML can be used in many ways to enhance the user experience, through use of sophisticated multi-level menus, displaying descriptions of what the user taps and holds, or sorting a table by the column that the user selects.

For guidance on using DHTML in M-Business Anywhere mobile applications, see "Using Dynamic HTML" (page 36). Additional information on the individual components of DHTML is presented in the following sections.

HTML 4.01

The M-Business Anywhere client supports HTML 4.01, including XHTML 1.0, and a portion of cascading style sheets (CSS) Level 1. The W3C specification for HTML 4.01 is available at:

http://www.w3.org/TR/html401/

Multiple forms on a page, a feature long available on desktop browsers, are supported by the M-Business Anywhere client, allowing small forms to be combined on one page instead of requiring a separate download for each.

Links to launch email, `mailto:` links, also are now available in the M-Business Anywhere client.

Note

On mobile devices using Microsoft's email client, a BCC parameter and all parameters following will be ignored. If your users' mobile environment includes the Microsoft email client, avoid the BCC parameter altogether, or at least place it at the very end of the parameter string for `mailto:`.

CSS 1.0

The M-Business Anywhere client supports inline CSS attributes. Embedded and external style sheets are not supported at this time. Support of the inline CSS attributes allows JavaScript to dynamically modify the formatting of HTML elements. See "Formatting: Using JavaScript to manipulate CSS attributes" (page 37).

DOM Level 1

The M-Business Anywhere client supports most features of DOM Level 1. These features allow JavaScript to dynamically create and modify HTML elements. See "Constructing: Using JavaScript with DOM to dynamically create and change page elements" (page 40).

JavaScript

TheM-Business Anywhere client includes the M-Business JavaScript engine, implementing a major subset of the Netscape JavaScript standard. JavaScript can manipulate CSS attributes, to dynamically change formatting of existing HTML elements, and call DOM methods, to dynamically create and modify page elements.

The M-Business JavaScript engine provides the standard scripting environment that web designers have come to rely on to enhance professional web pages. In the M-Business Anywhere client, JavaScript is the programming language that allows developers to create applications that are as feature rich as the best desktop applications. The M-Business JavaScript engine allows direct access to most of the M-Business client extension API (PODS), so that developers can use this API without having to write C code.

For guidance on using JavaScript with CSS and DOM to dynamically change page elements, see "Using Dynamic HTML" (page 36). For more information on using PODS with JavaScript, see "Accessing M-Business client extension API from JavaScript" (page 45).

XHTML 1.0

XHTML is a reformulation of HTML 4.01 in XML, with more rigorous requirements for HTML tag usage to produce well-formed documents. For example, tags must be entered only in lower case letters, end tags (such as `</p>`, `</h1>`, `</body>`) may never be omitted, and tags must be properly nested (`<b><i>`some text`</i></b>`, instead of `<b><i>`some text`</b></i>`).

The trade-off for the extra effort necessary to make your pages conform to XHTML rules is that such pages can be read and processed by XML-enabled software. At the same time, your pages will display correctly in the M-Business Anywhere client and popular desktop browsers.

For more information on XHTML, see the comprehensive reference on XHTML maintained by the W3C:

http://www.w3.org/TR/xhtml1/

M-Business client extension API

The M-Business client extension API (PODS) provides access to most of the internal components of the M-Business Anywhere client. Most of the API methods that control the look of the M-Business Anywhere client or navigate in its history list are directly accessible from M-Business JavaScript engine, with no need to write C code. Writing C code is always an option, but it is necessary only when you wish to speed up computation-intensive processing or access the underlying hardware.

For summary information on the M-Business client extension API, see “Using M-Business client extension API” (page 45).

Custom branding for M-Business Anywhere client

One of the easiest things to customize in the M-Business Anywhere client is its branding. Several features are available that allow enterprise administrators to custom brand the M-Business Anywhere client:

- You can configure M-Business Anywhere client to display a specific web page when launched.
- You can specify the error message to be displayed if the specified URL cannot be displayed.
- You can replace the standard M-Business Anywhere client icon on the mobile device with a custom icon.

For guidance using custom branding, see “Implementing custom M-Business Anywhere client branding” (page 44).

When to use the M-Business client extension API

The M-Business Anywhere client provides APIs that allow developers to access nearly all of the M-Business Anywhere client's internal components. The M-Business Anywhere client now is infinitely extensible, within the limits of the mobile device's capabilities and the practical constraint of the budget available for developers to write and test code.

Most of the APIs are directly available to M-Business JavaScript engine, which is an integral part of the M-Business Anywhere client. Because of JavaScript's relative ease of development, iAnywhere, Inc. recommends developing your applications in JavaScript if that environment meets your needs.

If your customizations involve high-volume number crunching, require hardware-level access to the mobile device, or require API functions that are not available to M-Business JavaScript engine, you should develop your customizations in C.

If your application requires any of the functionality listed below you can use the M-Business client extension API, accessed either from JavaScript or C, to complete your application:

- Check or wireless connection status; connect under application control.
- Manipulate the browser window.
- Control hardware.
- Vend documents.
- **U**se the Document Object Model (DOM).
- Control offline form submissions.
- Read screen attributes.
- Access a connected Symbol Technologies scanner.
- Initiate a sync with M-Business Server.
- Customize the M-Business Anywhere client toolbar.
- Read and change user preferences.

For guidelines on using the M-Business client extension API, see "Using M-Business client extension API" (page 45).

For detailed information on developing custom applications, in JavaScript or C, as well as guidance in determining which environment is best for a particular application, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, referenced in "Related publications" (page 8).

Security options

The M-Business Anywhere architecture allows system administrators to secure not only the information that is exchanged, but also the identity of users and their sponsoring networks. Such protection allows companies to deploy mobile applications that transmit highly sensitive information, such as financial and patient data.

M-Business Anywhere supports deployment of security technologies at three points in the M-Business Anywhere architecture: when the M-Business Anywhere client communicates with the M-Business Anywhere server, within the M-Business Anywhere server itself, and when the M-Business Anywhere server connects with a web server.

For guidance on implementing M-Business Anywhere security options, see “Implementing security options” (page 54).

For an overview of security options in the M-Business Anywhere architecture, see *Ensuring Mobile Security from the Device to the Datacenter*, referenced in “Related publications” (page 8).

For more information on enabling security on M-Business Server, see the *Administrator Guide for M-Business Server*, referenced in “Related publications” (page 8). For user instructions on enabling security on the M-Business Anywhere client, see the *User Guide for M-Business Client*, referenced in “Related publications” (page 8).

CHAPTER 3. Guidelines for designing good mobile applications

Implementing caching of web pages

For optimum performance, you should configure the M-Business Anywhere server and the M-Business Anywhere client on user mobile devices to cache web pages.

Implementing server-side caching

On the server side, caching must be enabled and configured on the M-Business Anywhere server. To make server-side caching as efficient as possible, all web servers within your company that support channels should be configured to supply appropriate caching information through HTTP headers.

Enabling and configuring the M-Business Server shared cache

Server-side caching is enabled by default when you install the M-Business Anywhere server. Table 3-1 lists some important parameters, stored in the *syncui.conf* file, which enable and control server-side caching on the M-Business Anywhere server.

You change these settings by editing the *syncui.conf.default* file, then running a setup script specific to the server operating system. For detailed instructions on setting these and other parameters that affect the server-side cache, see the *Administrator Guide for M-Business Server*, "Setting cache preferences" heading.

Table 3-1 Some important server-side cache settings

Setting	Description
`ServerSideCaching`	Controls whether server-side caching is enabled or not. The default setting is `On`.
`MaxCacheSize`	The maximum document size, in bytes, that the M-Business Anywhere server will try to cache. If you sync large documents, increasing this value to a value larger than your largest document will improve performance. The default is 262144 (256KB).
`CacheRoot`	The absolute path to the directory where cached files are to be stored.
`CacheSize`	The approximate size limit for the server-side cache. This is only approximate because the M-Business Anywhere server trims the cache only at intervals specified by `CacheGcInterval`, and the cache grows in between trimmings.

Setting	Description
`CacheGcInterval`	The cache's garbage collection interval in hours. During garbage collection, the M-Business Anywhere server trims the cache to the `CacheSize` parameter.

The most important setting is the `ServerSideCaching` parameter, which enables or disables server-side caching. `ServerSideCaching` is on by default; to ensure that server-side caching is still enabled, you should check that `ServerSideCaching` is still set to `On`.

Use the `CacheRoot` parameter to specify the location of the server-side cache. Performance will be best if the server-side cache is located on a hard drive attached directly to the same machine where the M-Business Anywhere server is installed.

The size of the server-side cache is controlled by the interaction of the `CacheSize` and `CacheGcInterval` parameters. Proceed as follows to set these parameters appropriately:

- Determine how much total disk space you want to devote to the server-side cache.
- Determine how often you want the M-Business Anywhere server to perform garbage collection. This will be the `CacheGcInterval`.
- Estimate how much the cache will grow in the interval between one garbage collection and the next.
- Subtract the estimated growth in the cache between one garbage collection and the next from the total disk space you want to devote to the server-side cache. This will be the `CacheSize`.

Monitor the server-side cache over time and make adjustments to these parameters as necessary.

Providing HTTP header directives on the web server

In order for caching to work most effectively, it is necessary for the web server supporting the channel to caching information for each page involved. The web server does this by adding special directives to the HTTP headers that it generates for a page.

The `Last-Modified` header specifies when the page was last modified. The M-Business Anywhere server can use this information to determine whether the page on the web server has been modified since the copy in the server-side cache was downloaded.

The `Cache-Control` header instructs the M-Business Anywhere server to cache the page for a specific period of time, so that the M-Business Anywhere server can safely serve the page from its cache without even checking with the web server.

There is also an `Expires` header that you could use in place of the `Cache-Control` header — it says, in effect, "cache this page until this

date." Although use of `Expires` headers is not recommended by iAnywhere, Inc., they are supported.

Note

The greatest savings obtained from caching tend to be realized from by just setting the `Cache-Control` header interval to something in the range of 20 minutes to one hour. To be sure that users do not receive an outdated page, it is best to set the `Cache-Control` header interval to 1/20th or less of the expected page update interval.

Table 3-2 Rough guidelines for setting `Cache-Control` header `max-age` times

Type of page	How often changed	`Cache-Control` header `max-age` setting
long-term article or reference	every 6 months, or less often	604800 (1 week)
front page with daily news	daily	3600 (1 hour)
stock index	every 20 minutes	60 (1 minute)
personalized generated page	--	setting appropriate for how often changed, but add `private` directive

The specific steps to follow in order to implement HTTP headers for web pages depend on how the pages are created and the web server supporting the site. Instructions for implementing `Cache-Control` headers with static pages on the Apache web server are available in the *M-Business Channel Developer Guide*, referenced in "Related publications" (page 18).

More information on server-side caching: For more detailed information on server-side caching, see the *Administrator Guide for M-Business Server*, "Setting cache preferences" heading for the appropriate server operating system, and the *M-Business Channel Developer Guide*, referenced in "Related publications" (page 18).

Implementing client-side caching

Configuring users' client online cache from the M-Business Anywhere server

While the user has ultimate control over the on-device client cache, you can configure client-side caching from the M-Business Anywhere server.

Table 3-3 Some important client-side cache settings

Setting	Description
`DefaultOnDevice Expiration`	The time in seconds after which a page cached on device expires, if no other setting is used. The default is 1200 seconds (20 minutes).
`OnDeviceCaching Option`	Controls on-device caching. The following values are valid: **`UseDefault`** - cache all documents for the amount of time indicated by the document's HTTP headers. If a document has no HTTP header caching information, cache it for the number of seconds set in `DefaultOnDeviceExpiration`. **`EverythingDefault`** - ignore any HTTP header caching information and use the `DefaultOnDeviceExpiration` value for non-error documents; do not cache error documents. **`EverythingAlways`** - cache all documents on the device indefinitely. **`NothingEver`** - never cache any document.

More information on server setup of client-side caching: For more detailed information on client-side caching, see the *Administrator Guide for M-Business Server*, "Setting cache preferences" heading for the appropriate server operating system.

User configuration of client online cache on the device

On Palm OS and Windows CE OS, users can set the maximum size for the client cache.

All users can clear their client caches entirely, or remove individual pages, in order to free up memory for other purposes.

More information on user control of client-side caching: For more information on user control of the client cache, see the *User Guide for M-Business Client*, "Managing online cache" heading for the appropriate mobile device operating system.

Using Dynamic HTML

Constraints

It is likely that you will do most of the creation and initial testing of the HTML pages for your mobile application on a desktop browser, such as Internet Explorer or Netscape Navigator. Because these browsers support a larger range of the HTML 4.01 feature set, it is important that you are aware of what is *not* supported by the M-Business Anywhere client.

For a list of features that the M-Business Anywhere client does and does *not* support from HTML 4.01, CSS 1.0 attribute tags, DOM level 1, and the Netscape JavaScript standard, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "Supported and unsupported features in third-party standards" heading.

Note

The M-Business Anywhere client has an upper limit of 64,535 DOM objects per HTML page. This is the total count of tags plus text objects. Most likely you will run into the memory limits of the mobile device before you reach this DOM object limit.

Maximum page size

The maximum page size that the M-Business Anywhere client can display is difficult to determine from the size of the page file in the desktop file system. The ultimate limit is the amount of available memory on the mobile device — after the memory required for the M-Business Anywhere client and page caching are factored out.

Because pages are compressed on the device, a page whose size on your desktop file system is larger than the available memory may easily fit. There is no firm compression ratio that you can use to estimate compressed page size from the page's size in your desktop file system.

If you have been dealing with very large pages in version 4.2, you should be aware that it is likely that the same pages, displayed on M-Business Anywhere client version 5.5, will take anywhere from 1.1 to 2.5 times as much memory on the device. There are two reasons for this: pages in version 5.5 contain layout information that was absent in version 4.2; and version 5.5 supports a larger range of HTML tags, so less unrecognized HTML can be stripped out before the server downloads the page to the M-Business Anywhere client.

Basic strategies

There are two basic strategies you can use in DHTML to create special effects on web pages, in response to user actions. Of course you can combine these two strategies as needed:

- **Formatting** - Using JavaScript to manipulate CSS attributes to control the formatting of HTML objects. This approach can be used when working with content that already exists, such as a static page

stored on the server. This approach can also be used with a dynamic page, generated at the time the page is requested, as long as the details of the HTML object(s) to be manipulated are predictable. For more details on this approach, see "Formatting: Using JavaScript to manipulate CSS attributes" (page 37).

- **Constructing** - Using JavaScript with DOM methods to dynamically create and modify page elements. This approach can be used when working with an unknown set of data, such as output from a query on a backend database. This approach is also effective when displayed data must be sorted in a different order or dynamically extended. For more details on this approach, see "Constructing: Using JavaScript with DOM to dynamically create and change page elements" (page 40).

Formatting: Using JavaScript to manipulate CSS attributes

The target portion of your page — the portion of the page that will have its displayed appearance changed, or be selectively hidden and displayed — may be any HTML element. You can target a group of HTML elements by bracketing them with `<div>...</div>` tags. You can target a segment of text within a larger HTML element by bracketing it with `<span>...</span>` tags.

The generalized steps involved in implementing the formatting approach are outlined below, using the simple example of a text segment within a paragraph that changes its background color from yellow to red when tapped.

- First write the HTML with all elements displayed.

```
<p>This text starts the paragraph.
This is target text within the paragraph.
This text ends the paragraph.
</p>
```

- If necessary, bracket the target portion of the page that will be selectively hidden and displayed, or have its displayed appearance changed, with `<span>...</span>` or `<div>...</div>` tags.

```
<p>This text starts the paragraph.
<span>
This is target text within the paragraph.
</span>
This text ends the paragraph.
</p>
```

Note

With any HTML element, you can specify the attributes directly in the element tag. You only need to add `<span>...</span>` or `<div>...</div>` tags when you want to target text within a larger HTML element (`<span>...</span>`) or when you want to target a group of HTML elements as a unit (`<div>...</div>`).

- If needed, add an `id` attribute to your target's opening tag to give it a name (which must be unique in the page):

```
<p>This text starts the paragraph.
<span id='my_id'>
This is target text within the paragraph.
</span>
This text ends the paragraph.
</p>
```

Note

You need to include an `id` attribute in the target HTML element when users will be tapping a different element to change the target element's formatting. You use the name defined in the `id` attribute to reference the target element from the element that is tapped. In this simple example, users will be tapping the target element directly, so the `id` attribute added above is not needed.

- Give the target its initial look, using `style` attributes:

```
<p>This text starts the paragraph.
<span style="background-color: yellow;">
This is target text within the paragraph.
</span>
This text ends the paragraph.
</p>
```

- Add an `onClick` attribute to the tag of the HTML element which users tap to change the appearance of the target.

```
<p>This text starts the paragraph.
<span style="background-color: yellow;" onClick=>
This is target text within the paragraph.
</span>
This text ends the paragraph.
</p>
```

Note

In this simple example, the HTML element that changes is the same one that users tap. Examples that follow illustrate tapping one HTML element to change the formatting of another.

- Add the JavaScript code to be executed by the `onClick` attribute.

  The JavaScript code can be placed immediately after the `onClick` attribute, but for readability should be defined as a function and placed in the page header, between `<script>...</script>` tags, if the code gets very long.

```
<p>This text starts the paragraph.
<span style="background-color: yellow;"
onClick="style.backgroundColor='red';">
This is target text within the paragraph.
</span>
This text ends the paragraph.
</p>
```

Note

Although `onClick` is the most commonly used event handler, you may also use a number of others, such as `onMouseDown` and `onMouseUp`. For more information, see the *Developer Guide for M-*

Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API, "HTML 4.01 tags and attributes, including CSS" heading.

Toggling an attribute

The JavaScript in the example above changes the background color of some text from yellow to red when the user taps the text. Often you want the user to be able to tap the same HTML object again to reverse, or toggle, the effect of the previous tap. In order to do this, your JavaScript code must determine the current state of the element by either reading the state each time or by storing the state in a variable. The example below toggles the target text's background color by reading the current background color each time:

```
<p>This text starts the paragraph.
<span id="my_id" style="background-color:yellow;"
onclick="
   if
(document.getElementById('my_id').style.backgroun
dColor=='yellow') {
document.getElementById('my_id').style.background
Color='red';
   } else {
document.getElementById('my_id').style.background
Color='yellow';
   }">
This is target text within the paragraph.
</span>This text ends the paragraph.
</p>
```

The following example toggles the target text's background color by storing a value in a variable, which is then toggled each time the background color is toggled.

```
<script>
   var toggle='false';
</script>
...
<p>This text starts the paragraph.
<span style="background-color:yellow;" onClick="
   if (toggle=='false') {
      style.backgroundColor='red';
      toggle='true';
   } else {
      style.backgroundColor='yellow';
      toggle='false';
   }">
This is target text within the paragraph.
</span>This text ends the paragraph.
</p>
```

Expanding and collapsing a list: Hiding and displaying elements

You can achieve many special effects with DHTML by selectively hiding and displaying HTML elements. One example is expanding and collapsing a hierarchical list, such as a menu of options.

```
<h3 onClick="
   if
(document.getElementById('my_para').style.display
=='none') {
document.getElementById('my_para').style.display=
'inline';
   } else {
document.getElementById('my_para').style.display=
'none';
   }">
Heading text
</h3>
<p id='my_para' style="display:none">
List Item #1<br>
List Item #2<br>
List Item #3<br>
</p>
```

For simplicity, the list items in the sample above are plain text. To turn the list into a drop-down menu, just replace the literal text with links.

For more examples illustrating JavaScript manipulation of formatting of HTML elements to build feature-rich, desktop-style mobile applications, see *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "JavaScript engine code samples" heading.

Constructing: Using JavaScript with DOM to dynamically create and change page elements

Using DOM to create or modify components of an HTML page tends to be more abstract than manipulating CSS attributes — you must keep track of aspects of the document's hierarchical object structure. The payoff for dealing with this extra complexity is that you gain complete scripting control over the page: Besides changing attributes of existing elements, you can create new elements.

Below is a simple example of sample code that adds a new list element to the bottom of a list each time the user taps a button. Boldface portions mark the DOM methods being called, plus the key `id` attribute that names the unordered list element so that it can be referenced by `getElementById()`.

```
<script>
    var counter = 1;
    function addListItem()
    {
        var ul = document.getElementById('ul_id');
```

```
        var new_li = document.createElement('li');
        var new_text = document.createTextNode('List
           Item #' + counter);
        ul.appendChild(new_li);
        new_li.appendChild(new_text);
        counter++;
    }
</script>
...
<body>
...
    <h1>Heading</h1>
    <input type="button" onClick="addListItem();"
           value="Add Item">
    <ul id="ul_id">
        <li>List Item #0</li>
    </ul>
    <p>
        This is a paragraph below the list.
    </p>
</body>
```

For more examples illustrating JavaScript construction of HTML elements to build feature-rich, desktop-style mobile applications, see *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "JavaScript engine code samples" heading.

Some general DHTML tips for M-Business Anywhere client pages

Below are some recommendations for working with DHTML in your mobile applications that will display pages on the M-Business Anywhere client:

- **Make use of `<span>` and `<div>` tags**: Remember that you can have an `onClick` event handler on any portion of an HTML page by surrounding the target with `<span>...</span>` tags, if the target is a text segment within another HTML element, or `<div>...</div>` tags if the target includes multiple HTML elments.
- **Simulate frames**: When you need frames, use a table layout to partition the screen as frames would. For example, you could create a one-row, two-column table with a narrow column on the left for a cell that contains a menu, and a wide column on the right for a cell that holds the main content. Then use JavaScript in the menu elements to manipulate the CSS attributes of the main content to selectively display and hide different portions.
- **First write HTML to display all page elements**: When you are creating a page which will involve hiding large parts of the content and then displaying it selectively under user control, start by writing the basic HTML that displays all the parts of the page. After you have gotten all the pieces looking right in this full-page display, you

then can add the CSS attributes to hide all the parts of the page that should not be part of the initial page display. Lastly, you can add the JavaScript code to manipulate the CSS attributes to display and hide the parts of the page in response to user interaction.

- **Be aware of desktop browser speed**: When developing pages on a desktop browser, remember that the same page on a mobile device will render significantly more slowly. If a page feature seems a little slow in a desktop browser, you will probably want to redesign it so that speed on a mobile device is acceptable.
- **Avoid `onMouseOver`**: The hardware on most mobile devices does not support the `onMouseOver` event handler, so it is best to simply avoid its use.
- **Use `getElementById()`**: When writing JavaScript to manipulate CSS attributes, use `getElementById()` to specify named elements, instead of `document.all`... . The M-Business Anywhere client is optimized for `getElementById()`.

General page design considerations

Whatever you do with DHTML, or plain HTML, the mobile device screen presents quite different design constraints compared with browsers on a desktop or laptop screen. The biggest constraint is the small screen; it makes the general design principle of avoiding horizontal scrolling more difficult to achieve.

- **Design for lowest common denominator**: If your user population is distributed across multiple mobile device platforms, make sure you design for the smallest screen, least number of colors, and least selection of fonts.
- **Avoid tables if you can**: As far as possible, stack tabular data into body text that sizes to fill the screen width, with longer text lines wrapping. Consider use of `<pre>` tag with columnar data aligned by use of spaces with the fixed-width font.
- **Keep necessary tables narrow**: When tables are unavoidable, keep down the number of columns, keep the columns narrow, and let columns dynamically resize to fill the screen width.
- **Combine multi-page series into one page**: When it is desirable to break a long page into a series of smaller pages, try to structure the material as a single page and use JavaScript and DHTML attributes to hide and display sections in sequence. With a series of separate pages, some later pages may not be downloaded because of the link depth setting for the channel.

For more detailed guidance on designing HTML pages for the M-Business Anywhere client, see the *M-Business Channel Developer Guide*, referenced in “Related publications” (page 18).

Implementing custom M-Business Anywhere client branding

In order to have the M-Business Anywhere client use a custom icon and display a custom message if the home page is unavailable, it is necessary to write a launcher application for the operating system. You can write such a launcher application using any language you choose that can produce a binary executable for the target platform. iAnywhere, Inc. Professional Services also offers this service.

You can customize the M-Business Anywhere client home page by either one of two methods:

- Modify the *index.html* page in the *device* directory on the the M-Business Anywhere server host machine.
- Change the home page URL in the mySQL database to point to your page located elsewhere.

Using M-Business client extension API

For detailed instructions on using M-Business client extension API to support your HTML pages, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API,* referenced in "Related publications" (page 18).

Accessing M-Business client extension API from JavaScript

You may make use of M-Business client extension API without even being aware of it. Major portions of M-Business JavaScript engine are implemented through this API, but the implementation is seamless; you do not need to do anything special to access the portions that are accessible to JavaScript.

All API interfaces documented in the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API* include separate JavaScript syntax entries for all attributes and methods that are available to M-Business JavaScript Engine. You may refer to these reference sections, or use a third-party reference materials where applicable.

If you are relying on third-party documentation, make sure you are aware of the features which are not supported by the M-Business Anywhere client. See *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "Unsupported features in third-party standards" heading.

Writing C code (PODs) to use M-Business client extension API

It is necessary to write C code only when you need access to the mobile device hardware, when you need the maximum processing speed the device can deliver, or when you need the full feature set of the M-Business client extension API. The list of features below cross references the PODS object, or group of PODS objects, which you would use to implement the feature, along with the section of the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API* where the PODS objects are documented.

- **Wireless connection status** - Determine whether a wireless device is connected and automatically connect a disconnected wireless device when a connection is needed. See the reference section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODSAvantGo object" heading.
- **Browser window** - Read window attributes and display messages, navigate and manipulate the history list, reload or replace the currently displayed page, parse the current URL. See the reference

section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODSWindow object" heading.

- **Hardware** - Control accessible hardware (only through C). See "Writing device drivers" headings in the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, under the "ADOMDOMImplementation object" and "PODSEventMgr object" headings.
- **Document vending** - Make a list of documents available, with capability of moving through the list. See the reference section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODS document-related objects" heading.
- **Offline form submissions** - Control how offline submissions behave and manipulate the queue of submitted forms. See the reference section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODS submission-related objects" heading.
- **Screen** - Read screen attributes. See the reference section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODSScreen object" heading.
- **Symbol Scanner** - Access a Symbol Technologies scanner connected to a Palm device and use the data to populate a form for the user to submit. See the reference section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODSSymbolScanner object" heading.
- **Sync** - Initiate a sync with M-Business Server. See the reference section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODSAvantGo object" heading.
- **Toolbar customization** - Modify the M-Business Anywhere client toolbar by adding your own buttons that perform custom operations. See the reference section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODSToolbar object" heading.
- **User preferences** - Read and change user preference settings. See the reference section of *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "PODSPrefs object" heading.

For detailed information on developing custom applications, in JavaScript or C, as well as guidance in determining which environment is best for a particular application, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*.

Making offline form submissions more user-friendly

Default behavior of offline submissions

Standard HTML forms are designed for online submission: When the user clicks the Submit button, the form data is sent directly to the server, which processes the data and sends back a response page.

When a user submits a standard HTML form in offline mode, the form is queued up for submission the next time the user syncs, and the user sees a generic message about the submission, followed by the form as it looked before data was entered. When the user syncs, the response page returned from the server is not automatically displayed in the M-Business Anywhere client; instead it is downloaded into the Form Manager, where the user has to remember to look for it.

New users in particular may get confused and re-enter the form data. Any user may forget to look for response pages in the Form Manager after syncing.

Using JavaScript to control offline form submission in the M-Business Anywhere client

Instead of using a standard HTML submit button, you can add a JavaScript `onClick` event handler that calls the JavaScript `submit` method with M-Business client extensions that let you fine-tune the behavior of the offline submission.

For example, if you have an order form that returns an order confirmation when processed on the server, the standard HTML for a `submit` button would look something like this:

```
<input type="submit" name="submit" value="Submit">
```

To avoid the problems noted when the form is submitted offline, in place of this standard HTML, use the JavaScript `submit` method with M-Business client extensions:

```
<input type="button" name="submit" value="Submit"
onClick="form.agsubmitDiscardResponse = true;
form.agsubmitDisplayDefaultMessage = false;
form.agsubmitHidden = true;
form.agsubmitMessage = 'Customer order will be
           submitted next time you sync; Look for
           confirmation in your Confirmation
           channel';
document.forms[0].submit;
back()"
>
```

Note When the user submits the form containing the JavaScript above in online mode, the M-Business Anywhere client ignores the M-Business client extensions and sends the form data directly to the M-Business Anywhere server.

In the above example, here is what the bolded items do in offline mode:

- **`agsubmitDiscardResponse`** - Setting this M-Business Anywhere client attribute to `true` deletes the response.
- **`agsubmitDisplayDefaultMessage`** - Setting this M-Business Anywhere client attribute to `false` suppresses the display of the default submission message.
- **`agsubmitMessage`** - This M-Business Anywhere client attribute is used to specify the text of the message to use in place of the default submission message.
- **`agsubmitHidden`** - Setting this M-Business Anywhere client attribute to `true` hides the submission in the Forms Manager.
- **`submit`** - This JavaScript method places the form submission into the Form Manager's queue, to be sent to the server during the next sync.
- **`back()`** - This JavaScript method displays the page before the form page in the M-Business Anywhere client's history list. The default here would be to redisplay the form just submitted, in its initial state.

For more information on these and other M-Business client extensions for the JavaScript `form` object, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, "Form object extensions" heading.

Using cookies to support personal channels

A cookie is used to store information about the browser user, so that a web server can personalize content sent to that user. You can use cookies with the M-Business Anywhere client the same way you would with other browsers, but you also can use them to support personal channels.

- Create a cookie for the user that contains a unique user ID,
- Maintain a database on the web server that contains a table of user IDs, plus any additional profile information that you need to use to support personal channels.
- If you will need to collect some of the information for the cookie from the user, then implement a form that allows users to use to specify this information and stores it in the database.

Note

You could implement the personal channels without using cookies, but that would require users to enter an identifier in each form submitted.

For information about managing cookies on the M-Business Anywhere server, see the *Administrator Guide for M-Business Server*.

Using personal channels to organize response pages

The best way to make the response pages for multiple offline form submissions available to the user in a well organized, user-friendly way is to create a personal channel for each user and pass the response pages through that channel.

To support personal channels, have the script receiving the form write each response page to a disk file in a user-specific directory. Derive the name of that directory from the user's cookie. At the same time, have the script generate an index page in the same directory. The index page should link to all the response pages and must have a fixed file name.

Having done that, you can set up personal channels for each user of your application. The index page and response pages will then be updated on each user's device at the end of the same sync in which the form data submitted offline is sent to the server.

Note

The M-Business Anywhere server ensures that processing of the offline form data is completed before downloading any channel pages to the user. Thus any changes in channel pages that result from the offline form submissions in a sync will be reflected in affected channel pages that are downloaded in the same sync.

The dynamically generated index page is the top level URL for the channel. The link depth for the channel should be set to one.

For instructions on setting up a personal channel, see the *Administrator Guide for M-Business Server*, "Creating a personal channel for a user" heading.

Using M-Business Anywhere client HTTP request headers to customize content

Whenever the M-Business Anywhere client sends a page request to a web server, it includes several headers that can be used by the receiving process on the server to determine characteristics of the user's hardware and connection. The process can then branch on that information and download, or dynamically generate, different pages that are optimized for different devices, or different pages for online vs. offline modes.

Table 3-4 M-Business Anywhere client HTTP request headers

Header	Description	Base 64?*	Typical Values
`User-Agent`	The typical User-Agent string that's sent with any browser request	No	`Mozilla/3.0 (compatible; AvantGo 3.2)`
`X-AvantGo-Screensize`	The size of the screen of the handheld device, in pixels	Yes	`150x150` `240x320`
`X-AvantGo-DeviceOS`	The Operating System of the device that is accessing the page	Yes	`PALM_OS` `WINCE_OS`

`X-AvantGo-ColorDepth`	The bit depth of the handheld device that is viewing your page	Yes	`2`
			`4`
			`8`

`X-AvantGo-OnlineRequest`	Is the user browsing in online mode, with a live connection?	No	`TRUE` - In offline mode, header is simply omitted.
`X-AvantGo-SecureSync`	Is the user browsing with a secure connection?	No	`TRUE` - In offline mode, header is simply omitted.
`X-AvantGo-Version`	What version of the client does the user have?	No	`3.3.350`
*"Base64?" means "Base64-encoded?"			

Different processes that dynamically creating web pages retrieve the request header information in different forms. Also, the base64-encoded items must be decoded before your server-side process can use the values. Consult the documentation for the server process that generates the web pages for guidance on using M-Business Anywhere client request headers to customize content.

For examples of ways in which content can be customized using M-Business Anywhere client request headers, see the *M-Business Channel Developer Guide*, referenced in "Related publications" (page 18).

Using URL macros to customize content

URL macros on the M-Business Anywhere server, listed in Table 3-5 (page 52), allow you to set up a single channel that delivers different content to different users. The content differences can be based on the user name, the user's device, or even the processor on the user's device.

You use URL macros by inserting them into some portion of the channel URL (**Location** field on the **New Channel** page) when you define a channel. M-Business Anywhere client HTTP request headers supply M-Business Server with the specific values with which to replace the URL macro in the channel when M-Business Server receives an M-Business Anywhere client sync request.

Note

You must ensure that content exists for all the specific URLs that will result when when users sync and URL macros in the channel URL are replaced by specific values.

Below are some simplified examples of mapping the AG_USER URL macro to web server directories and files. In these examples, the it is assumed that the web server's domain is `http://myserver.com`, that this domain maps to the file system path *D:\webserver\contentroot*, and that you have three user IDs: `aasmith`, `bjwong`, and `ccgomez`.

- Different files for different users, in same directory:

  Channel URL - `http://myserver.com/myapp/AG_USER.asp`

  Files for different users -
  D:\webserver\contentroot\myapp\aasmith.htm
  D:\webserver\contentroot\myapp\bjwong.htm
  D:\webserver\contentroot\myapp\ccgomez.htm

- Different directories for different users, with same file name:

  Channel URL - `http://myserver.com/AG_USER/myapp.htm`

  Files for different users -
  D:\webserver\contentroot\aasmith\myapp.htm
  D:\webserver\contentroot\bjwong\myapp.htm
  D:\webserver\contentroot\ccgomez\myapp.htm

- User ID passed as parameter to web application:

  Channel URL - `http://myserver.com/myapp.asp?id=AG_USER`

  Files for different users -
  D:\webserver\contentroot\myapp.asp
  (The same file, *myapp.asp*, generates different content when each user ID replaces AG_USER in the channel URL.)

Table 3-5 M-Business Anywhere server URL macros

Macro	Description	Typical Values
AG_DEVICEOS	The Operating System of the device sending the sync request. Useful to optimize content for different device parameters.	WINCE_OS PALM_OS RIM_OS:MOBITEX RIM_OS:DATATAC
AG_DEVICEPROCESSOR	The processor of the device sending the sync request. Useful when downloading binaries through your channel to Pocket PC devices.	(Only applies when AG_DEVICEOS is WINCE_OS; blank for all other OS values) MIPS SH3 SH4 ARM
AG_USER	The User ID for the device sending the sync request. Useful whenever content needs to be user-specific.	User IDs set up on M-Business Anywhere server.

Designing applications to work both online and offline

An M-Business Anywhere client application that is well-designed for operation in offline mode will work just fine when used in online mode. The minimal requirement for an offline mode application is that each step in processing must not depend on getting back from the server the results from an earlier step. For example, a salesperson entering customer orders must not be required to wait for an order confirmation from the server before entering the next order.

An optional but highly recommended design standard for offline applications is to deliver the response pages for offline form submissions via personal channels. If the same application is used in online mode, the personal channel is ignored and the response pages automatically are returned directly to the M-Business Anywhere client after submission. For more information on, see “Making offline form submissions more user-friendly” (page 47).

If you wish to augment the M-Business Anywhere client’s differential handling of online vs. offline form submissions, the M-Business client extension API provides methods that can determine if a device is online, thus permitting code to branch and do different processing depending whether the device is operating in online or offline mode. However, you do not need to write any special code to make an application work seamlessly in both modes, unless there is a compelling logical reason for doing different processing based on the mode. If you want an application to work the same online or offline, all you have to do is be sure that the application works the way you want in offline mode.

For information on using M-Business Anywhere client request headers to determine whether a page request was submitted in online or offline mode, see “Using M-Business Anywhere client HTTP request headers to customize content” (page 50).

For information on the API methods that determine a device’s connection status, as well as methods that connect and disconnect a wireless device, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*.

For guidance on designing applications, as well as tips for designing web pages that display well on different devices, see the *M-Business Channel Developer Guide*, referenced in “Related publications” (page 18).

For details on implementing customizations in either JavaScript or C, and for more information on the trade-offs involved in choosing between these two development environments, see the *Developer Guide for M-Business JavaScript Engine, M-Business Client Extension API, and M-Business Database API*, referenced in “Related publications” (page 18).

Implementing security options

Securing M-Business Anywhere client-M-Business Anywhere server sessions

You can secure sessions between the M-Business Anywhere client and the M-Business Anywhere server by implementing industry-standard 128-bit SSL (Secure Sockets Layer) protocol (version 3.0). Unless you implement SSL through Microsoft Internet Explorer proxy settings, SSL is implemented through M-Business Anywhere client's default proxy settings.

SSL requires M-Business Server to have a digital certificate issued by a third party Certificate Authority. You can use an ECC type certificate, available from iAnywhere, Inc., unless you are implementing SSL through Microsoft Internet Explorer proxy settings. The latter requires an RSA type certificate.

Securing the M-Business Anywhere server

The M-Business Anywhere server automatically provides some significant security features. For example, the M-Business Anywhere server stores user cookies encrypted, using a 128-bit key. And when SSL has been enabled, the M-Business Anywhere server automatically performs transmission checks, monitoring all transmissions received to ensure that the SSL protocol is not being bypassed.

To make the M-Business Anywhere server and its communications with web servers and the M-Business Anywhere client as secure as possible, take all the standard precautions that security experts recommend for any server software that connects to the Internet.

Securing the M-Business Anywhere server-web server connection

In addition to enabling SSL, the M-Business Anywhere client-M-Business Anywhere server connection, you also can enable SSL for the M-Business Anywhere server-web server connection. SSL between the M-Business Anywhere server and web servers is used to authenticate secure pages.

SSL comes into play between the M-Business Anywhere server and a web server when a user has requested a secure page. The M-Business Anywhere server receives a digital certificate from the website. The M-Business Anywhere server checks the root certificate in the website's digital certificate against a file of trusted certificate authorities. The M-Business Anywhere server relays the page on to the user only if the website's root certificate is from one of these trusted certificate authorities.

For instructions on implementing SSL between the M-Business Anywhere client and the M-Business Anywhere server, and between the M-Business Anywhere server and web servers, see the *Administrator Guide for M-Business Server*, security chapter for the operating system under

which your M-Business Server is installed: “Security on Windows” or “Security on Solaris.”

For more background on security issues in the M-Business Anywhere architecture, see *Ensuring Mobile Security from the Device to the Datacenter*, referenced in “Related publications” (page 8).

Testing and deploying your mobile application

Most of the guidelines for testing and deploying any web-based application apply equally to mobile applications using the M-Business Anywhere architecture. This section covers only issues that are specific to the M-Business Anywhere environment.

Testing

Initial application testing can be done in a desktop browser if M-Business Anywhere client extensions are not involved (client extension API, certain JavaScript engine features). Even if your application will use M-Business Anywhere client extensions, you may find it more convenient to do initial testing of the HTML and most of the JavaScript pieces in a desktop browser.

If you are developing an application for the Palm OS, initial testing can be done on the desktop in the Palm emulator software.

If you are developing an application that will be used on RIM OS, in addition to other platforms, it is highly recommended that you do initial development and testing on the RIM platform.

When you begin testing application components on a mobile device, you may be able to save time by copying files directly to the device, then opening HTML pages through the **Open Page** dialog box. This may be faster than running the files through a channel and syncing the mobile device.

Setting up users and groups

Unless your application is intended for everyone in the company to use, you probably have in mind particular individuals or groups as the application's target audience. All the users for your application must be defined on your M-Business Anywhere server individually. For convenience, individual users can be assigned to groups so that the M-Business Anywhere server system administrator can provide the whole group access to the same content.

Groups are probably already defined on your M-Business Anywhere server for functional workgroups, such as sales, human resources, and top management. There may even be a group for everyone in the company who uses a mobile device. You may be able to use some combination of existing groups to give users access to your application, or you may need to define a new group.

For instructions on setting up users and groups, see the *Administrator Guide for M-Business Server*, "Managing users" and "Managing groups" chapters.

Setting up a channel and subscribing users

Your application will be delivered to the mobile devices of the groups of users that you specify through an channel. For an overview of how channels work, see “Understanding channels” (page 20).

When you set up a channel on the M-Business Anywhere server, you specify a single URL for the top level page of the channel content, and the link depth — the number of links away from that page for which pages should also be downloaded. For a graphic explanation of link depth, “Setting link depth” (page 20).

When you set up a channel, you also specify the channel audience. Channels can be categorized according to the audience targeted, as follows:

- **Group channels**: You specify one or more groups, and all the members of those groups are automatically subscribed to the channel.
- **Personal channels**: You specify a single user, and that user only is automatically subscribed to the channel.
- **Public channels**: By defining a channel as public, you allow any user to subscribe to it, but no one will be subscribed to it automatically. The channel will be listed in the M-Business Server’s list of public channels and individual users may subscribe to them through the M-Business Server desktop user interface or directly from the mobile device.

For instructions on setting up these different categories of channels, see the *Administrator Guide for M-Business Server*, “Creating channels for a group” heading, “Creating a personal channel for a user” heading, and “Using public channel publishing and administration” heading. For user instructions on self-subscribing to channels, see the *User Guide for M-Business Client*.

Index

H

J

L

M

O

P

R

S

T

U

W

X