Prior to subscribing for, installing or using any Third Party Products and Services it is your responsibility to ensure that your airtime service provider has agreed to support all of their features. Some airtime service providers may not offer Internet browsing functionality with a subscription to BlackBerry® Internet Service. Check with your service provider for availability, roaming arrangements, service plans and features. Installation or use of Third Party Products and Services with RIM’s products and services may require one or more patent, trademark, copyright or other licenses in order to avoid infringement or violation of third party rights. You are solely responsible for determining whether to use, Third Party Products and Services and if any third party licenses are required to do so. If required you are responsible for acquiring them. You should not install or use Third Party Products and Services until all necessary licenses have been acquired. Any Third Party Products and Services that are provided with RIM’s products and services are provided as a convenience to you and are provided “AS IS” with no express or implied conditions, endorsements, guarantees, representations or warranties of any kind by RIM and RIM assumes no liability whatsoever, in relation thereto. Your use of Third Party Products and Services shall be governed by and subject to you agreeing to the terms of separate licenses and other agreements applicable thereto with third parties, except to the extent expressly covered by a license or other agreement with RIM.

Certain features outlined in this documentation require a minimum version of BlackBerry® Enterprise Server, BlackBerry® Desktop Software, BlackBerry® Device Software and/or BlackBerry Handheld Software and may require additional development or Third Party Products and Services for access to corporate applications.

Certain products mentioned in this documentation include software developed by the Apache Software Foundation (http://www.apache.org/) and/or licensed pursuant to Apache License, Version 2.0 (http://www.apache.org/licenses/). For more information, see the NOTICE.txt file included with the software. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The terms of use of any RIM product or service are set out in a separate license or other agreement with RIM applicable thereto. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS OR WARRANTIES PROVIDED BY RIM FOR PORTIONS OF ANY RIM PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION.
Chapter 4
Optimizing web content for mobile device browsers

Objectives

- Describe techniques for designing efficient web content
- Describe how to use the profiling tools available in the BlackBerry® Web Plug-in for Eclipse and the BlackBerry® Web Plug-in for Microsoft® Visual Studio®
- Describe the use of offline form queuing
- Describe how the BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing network gateways use optimization techniques

This chapter outlines the principles for designing efficient web content for a BlackBerry® smartphone. It explains how to optimize source code using profiling tools, and how to use offline form queueing for greater efficiency. This chapter also describes how the BlackBerry® MDS Connection Service and the BlackBerry Internet Service Browsing network gateways use optimization techniques unique to the BlackBerry® Infrastructure.
Design efficient web content

On a BlackBerry smartphone, many factors—such as memory, screen size, and color depth—influence how content is rendered. When you create web content for a mobile browser, you can provide a better and more user-friendly wireless experience if you follow the basic design principles described in this section.

Follow basic principles of web design

Many standard principles of web site design apply when you create content for BlackBerry smartphones. Consider the following design recommendations when you plan your web site:

• Understand your audience—Determine who will use the site and the primary service that your site will provide. Mobile web users usually have different needs than computer users do. For example, mobile device users need to stay connected and need to access information and applications at anytime, and from any location. Such users are often working with time-sensitive information or performing urgent tasks. In addition, mobile device users are looking for a more personalized experience than computer users.

• Create an appropriate site hierarchy—Structure your web site based on its purpose, and organize the site to minimize the time that it takes users to find information or perform tasks.

• Provide useful links—Minimize the number of pages that users must navigate to accomplish their goals and consider the following guidelines for links:
  • Include a link to the home page on each page.
  • Whenever possible, include links to other related pages on your site, to minimize backward navigation using the browser history.
  • Categorize and group links.
  • Provide a clear label for each link.

Organize content effectively

Consider the following guidelines when you plan your web site:

• Deliver related content on as few pages as possible—Although a page with more content might take a few seconds longer to download, users do not have to make subsequent requests, and the information is available even when users move outside a wireless network coverage area. BlackBerry smartphone users can use the trackwheel to scroll through several screens of text easily. For example, on WML pages, put related cards in the same deck whenever possible so that the document has to be loaded only once. If the deck contains a relatively large card that many users might not want to view, save the card in its own deck to minimize download time.

• Stay focused on users’ immediate task—Simplify data selection and presentation to display only the information that users need at any one moment.

• Add links to related content—if you divide related content into more than one page, make links to related content easily accessible. Make sure that links to related content are visible in a non-
scrolling area of the page or at the top of the page. For example, you could add a More menu item (or soft key) to enable users to retrieve related content quickly.

- **Provide clear entry points**—Guide users by placing the most important tasks at top levels.
- **Design an effective layout**—Use a one-column layout with a flexible width so that it fits the screen, and ensure that the UI element size is readable without the need for the user to zoom in or out. Avoid horizontal scrolling; users are more tolerant of vertical scrolling.
- **Create logical blocks of information**—Use structural HTML markup to create logical blocks of information in your HTML. For example, use `<div>` elements to denote content blocks. Use heading elements (`<h1>`, `<h2>`, and so on) to define informational hierarchies.
- **Use cascading style sheets**—Use CSS to control web page layout and content presentation.
- **Create separate files for content, presentation, and interaction components**—External CSS and script files can be cached and reused the next time the user visits the web page, and reused the next time the user visits other web pages from your web site.
- **Avoid inline or internal styles**—Inline or internal styles and scripts are much less efficient, because, over time, they result in more content being transferred. Each time the user requests that page, the style and script content must come with it.

### Consider the BlackBerry smartphone screen size

Design web pages to use the BlackBerry smartphone screen effectively. BlackBerry smartphones have larger screens than most other mobile devices, such as mobile phones. Depending on the smartphone type and selected font size, the browser can display 12 to 18 lines of text with 28 to 35 characters on each line. In contrast, many mobile phone browsers display 4 to 7 lines of text, with 10 to 15 characters on each line.
Select the most appropriate markup language

When you create a new web site, you must decide whether you are going to write the source in HTML, WML, or SVG. Consider the following advantages and disadvantages of each markup language when you make your decision.

<table>
<thead>
<tr>
<th>Markup</th>
<th>HTML/XHTML</th>
<th>WML</th>
<th>SVG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td><strong>HTML can be migrated to XHTML much more easily than to WML</strong></td>
<td><strong>Most users of wireless web sites are accustomed to WML</strong></td>
<td><strong>Enables content developers to add movement and sound to their content</strong></td>
</tr>
<tr>
<td></td>
<td><strong>XHTML supports greater layout versatility than WML</strong></td>
<td><strong>Currently the most widely used markup language for wireless web applications</strong></td>
<td><strong>Offers dynamic layout and presentation support</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Functionality can be extended considerably using JavaScript®</strong></td>
<td><strong>Has a well-maintained DTD and is well-documented and supported</strong></td>
<td><strong>Automatically transcoded to PME format by the BlackBerry MDS Connection Service or BlackBerry Internet Service</strong></td>
</tr>
<tr>
<td></td>
<td><strong>XHTML has greater potential for future use, it will become the standard for mobile devices</strong></td>
<td><strong>Functionality can be extended using WMLScript</strong></td>
<td><strong>—</strong></td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td><strong>Usually larger than WML content, so it can take longer to display</strong></td>
<td><strong>Supports only basic page layout; best suited to very basic sites</strong></td>
<td><strong>SVG is not supported by the browser directly; it must be transcoded to PME format (either by BlackBerry MDS or BlackBerry Internet Service Browsing, or by the content developer)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Few XHTML resources are currently available</strong></td>
<td><strong>WMLScript is much less robust than JavaScript</strong></td>
<td><strong>Takes longer to download than other formats</strong></td>
</tr>
</tbody>
</table>
Use meta values to control the presentation of content

If you are writing your web code in HTML, you can use the `<meta>` element to provide additional information about the document. The BlackBerry® Browser supports the HandheldFriendly and viewport meta values to help you to control the presentation of your content on the BlackBerry smartphone screen.

On BlackBerry smartphones with BlackBerry® Device Software version 4.6 or later, by default, the BlackBerry Browser renders desktop-oriented content as it appears on a computer, then scales the content to fit the BlackBerry smartphone screen. For most desktop-oriented content, the BlackBerry Browser renders it at 20% to 40% of its original size.

Content that you design for the BlackBerry Browser should be presented to the user unscaled. You must inform the BlackBerry Browser that content is designed for the BlackBerry device screen by adding a `<meta>` tag to control the initial scale of your web page. Otherwise, the BlackBerry Browser scales your content in error.

The BlackBerry Browser supports the following methods to control the initial display scale with the `<meta>` tag:

- **HandheldFriendly**: The `HandheldFriendly` processing instruction informs the BlackBerry Browser that the content contained within the document is designed for small screens.
- **viewport**: The `viewport` definition allows you to specify the specific dimensions of the intended viewport.

Although there are differences in the implementation of each method, you can achieve the same result using either method.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Description</th>
<th>Availability</th>
</tr>
</thead>
</table>
| content       | text  | Required. This attribute sets the meta information for the name or http-equiv attributes. The value of the content attribute depends on the value of the following http-equiv or name attributes:  
• If http-equiv="refresh", this attribute specifies the URL to which the browser is redirected.  
• If http-equiv=" expires", this attribute specifies the number of seconds after which the page is cleared from the cache. A value of -1 indicates that the page is not cached, so that the browser requests and downloads the page every time it is accessed.  
• If http-equiv="cachecontrol", this attribute specifies how the page is cached. The following values are acceptable:  
  • Public: Page can be stored in public shared caches.  
  • Private: Page can be stored only in private caches.  
  • No-cache: Page cannot be cached.  
  • No-store: Page can be cached but not archived.  
• If name="HandheldFriendly", this attribute specifies whether or not the content is designed for small-screen handheld devices. Acceptable values are True or False.  
• If name="viewport", this attribute specifies the dimensions of the viewport.  
• The value may be one or more comma-separated name-value pairs. The BlackBerry Browser supports the width, userscalable, and initialscale viewport properties. | version 3.7 or later |
Chapter 4

The following example shows how to use the `<meta>` element to prevent unwanted content scaling.

```html
<html>
<head>
  
  <title>Test Page</title>

  <meta name="HandheldFriendly" content="true" />

</head>
```
Create content that adapts to different BlackBerry smartphones

Unless you intend to target a particular BlackBerry device, your content should be adaptable to all BlackBerry smartphones. You should create content that supports the multiple screen resolutions and sizes, and the different input and navigation methods available on BlackBerry smartphones. When you create a BlackBerry widget, it is important to note how BlackBerry widget development differs from BlackBerry Browser development for creating adaptable content. For example, a BlackBerry widget does not support zooming, or Column View. Consider the following guidelines:

<table>
<thead>
<tr>
<th>Best practice</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use CSS styles to control presentation wherever possible.</td>
<td>To target specific smartphones, it is easier to create a single HTML file with multiple CSS style sheets than it is to create multiple HTML files with similar content but different formatting and layout information.</td>
</tr>
</tbody>
</table>
| Create pages that are viewable on wide and narrow BlackBerry smartphone screens. | If necessary, consider creating separate style sheets for narrow-screen devices (BlackBerry® Pearl™), and wide-screen devices (BlackBerry® Curve™, BlackBerry® Bold™, and BlackBerry® Storm™). If you properly separate the presentation information from the HTML content, adapting to different screen sizes is simpler. You can also make your content adaptable to multiple screen sizes by using percentage-based measures. The following example demonstrates how to use the <meta> tag to disable automatic scaling and set the width and height of the page to the dimensions of the BlackBerry smartphone screen.  

```html
<meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0" />
```

When you set the page width and height to the dimensions of the BlackBerry smartphone screen, the pixels on the page match the resolution of the BlackBerry smartphone screen. |
Create pages that can resize properly when the orientation changes on a BlackBerry smartphone that has a touch screen.

Though percentage-based widths work well when transitioning between the portrait orientation and landscape orientation, percentage-based heights do not, particularly when using fixed-size text. To set the height of the page to work with different screen orientations, in a CSS, use the `<body>` element to fill the page. You should create a CSS for each orientation type.

The following example demonstrates how to use JavaScript to identify the orientation of the screen, and then set the height of the page using the appropriate CSS:

```javascript
function resetHeight() {
    document.getElementById("main-Body").style.height = screen.height + "px";
}

<body id="mainBody"
onload="javascript:resetHeight();" >

The following example demonstrates how to use JavaScript to reset the page height when the orientation changes:

```javascript
window.onorientationchange = function() {
    document.getElementById("main-Body").style.height = screen.height + "px";
}
```

When an orientation change resets the page height, this triggers a page update that resets the width as well.

Design content that is not dependant on one input method.

In most cases, the BlackBerry widget translates different user events, such as a trackball click or a touch gesture, to standard mouse or keyboard events. However, you should test your JavaScript code to verify that UI events are interpreted as expected on different devices.
Detect BlackBerry device information

Detecting device information can be a useful practice. There are instances in which it is helpful to have information about the requesting BlackBerry smartphone and version of the BlackBerry Device Software it is running.

When the BlackBerry widget makes an HTTP request, it includes the Profile header. This header provides the URL for the location of the User Agent Profile for the BlackBerry smartphone making the request. This URL uses the following form:

http://www.blackberry.net/go/mobile/profiles/uaprof/ <BlackBerry-model>/ <software-version>.rdf

In the form, the variables are defined as follows:

- `<BlackBerry-model>` is the BlackBerry smartphone model number, for example, 8220 or 9000.
- `<software-version>` is the BlackBerry Device Software version, for example, 4.3.1 or 4.6.0.

You can create a script to extract the Profile header information and let the content server return device-appropriate content in the HTTP response.

Encourage text entry

BlackBerry smartphone users can use the keyboard to type text into web forms. The browser supports both `<input type="text">` and `<textarea>` elements in HTML, and `<input type="text">` in WML.

Minimize download time

When you create content to be delivered over the wireless network, keep the following two key concerns in mind:

- Avoid increased data transfer costs. Many users must pay for the data transferred over wireless networks. Minimizing the expense of using your content should be one of your design goals.
- Be aware of reduced network bandwidth. Network latency is one of the common frustrations for content developers and users alike. Wireless networks cannot transfer data as fast as wired connections, so the more content you send, the longer the user must wait before the content is downloaded and rendered. With efficient content and smart delivery, you can minimize the impact of network latency.

Download time is affected by the following factors:

- content size
- wireless network
- protocol characteristics
For example, a 15-KB file can take 30 seconds or more to download through a WAP gateway on a GPRS network. You can improve download time by implementing the following practices:

- Reduce the size of web pages—To reduce the size of web pages, avoid unnecessary content and images. Reduce image file sizes as much as possible.
- Use scripting techniques to progressively render a page—Consider using AJAX scripting techniques to progressively render a page. Progressive page rendering refers to loading web page content in phases. You can reduce wait times for the user by sending a portion of the content to the BlackBerry smartphone, and then using the XMLHttpRequest object to request and receive additional content after the page has rendered. These requests occur in the background, so they do not block the user from viewing or working with the page.
- Push content proactively—Design web applications to push content proactively. When you push content proactively, the content is sent to the BlackBerry smartphone or browser cache without a user requesting and waiting for it.

**Improve rendering time**

Rendering on the browser does not affect the time it takes to display content as much as the download time does, but large content can still require several seconds to parse and display. To improve rendering times, perform the following actions:

- Create content using WML where possible. WML content typically is rendered more quickly than HTML or XHTML content.
- Process and filter content at an intermediate server between the web server and the smartphone.

The BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing will speed up rendering times by processing HTML content before sending it to the browser. These components filter out unsupported elements and convert content into a tokenized format that the browser can display efficiently.

**Create effective images**

Consider the following guidelines when you include images on your pages:

- Fonts that are saved as images should not be anti-aliased. Anti-aliasing smooths edges by blending the background and foreground colors. Anti-aliased images do not display optimally on the BlackBerry smartphone.
- If you resize an image to better fit the smaller screen, when possible, redraw the image. Scaling down the image results in blurred edges that display poorly.
- Although the BlackBerry MDS Connection Service and BlackBerry Internet Service Browsing can dither color images to monochrome, ideally your images should be saved in monochrome.
format for display on monochrome smartphones. The following example demonstrates examples of the same page rendered on a color and monochrome smartphone.

- If it is not possible to provide both a color and a monochrome image, verify that the image displays acceptably on both smartphone types.
- Users can specify whether images are loaded or not; therefore, images should not be critical to the effectiveness and usefulness of your web site.

Create effective monochrome images

The following examples demonstrate monochrome images that display well on the BlackBerry smartphone, and images that display poorly. The first pair of images display well because they are monochrome and contain well-defined edges. The second pair of images display poorly because of feathered edges and blurred colors.

![Figure 6.3 Examples of monochrome images that display well in the browser](image1)

![Figure 6.4 Examples of images that display poorly in the browser](image2)

To convert an image to monochrome using Adobe® Photoshop® 6.0, convert your image to a bitmap using the 50% threshold method. You may need to discard any color information by converting the image
to grayscale. The following diagram demonstrates how a gradient appears on the BlackBerry monochrome smartphones.

Gradients display clearly on the BlackBerry smartphone screen, but they are less effective on a smaller scale. For example, a font with feathered edges does not display clearly on the smartphone screen.

*Figure 6.5 Examples of grayscale gradients on a BlackBerry monochrome smartphone*
1. List two ways that you can use links to make web content efficient for BlackBerry smartphone users.

2. How many lines of text can a BlackBerry smartphone display?

3. List the three markup languages that you can use to develop web content for BlackBerry smartphones.

4. What is the purpose of the `handheldfriendly` meta value?

5. What is the purpose of the `viewport` meta value?

6. List three factors that affect download time.

7. Which type of content is rendered most quickly? Choose one.
   A. WML
   B. HTML
   C. XHTML

8. Fonts should be anti-aliased. True or false?
Answers

1. Link to the homepage on each page; link to related pages to minimize backward navigation.

2. 12 to 18

3. HTML
   WML
   SVG

4. The HandheldFriendly processing instruction informs the BlackBerry Browser that the content contained within the document is designed for small screens.

5. The viewport definition allows you to specify the specific dimensions of the intended viewport.

6. content size
   wireless network
   protocol characteristics

7. WML

8. False
Optimize source code using the BlackBerry Web Plug-in profiling tools

You can optimize source code by using the profiling tools that are included in the two BlackBerry Web Plug-ins that are available: the BlackBerry Web Plug-in for Eclipse, and the BlackBerry Web Plug-in for Microsoft Visual Studio. These profiling tools provide you with the ability to isolate and address coding bottlenecks.

You can use these profiling tools to profile the efficiency of code sections by setting a breakpoint at the start and end of the section of code that you want to profile. After you set the breakpoints, you can start a debugging session with the BlackBerry Smartphone Simulator and view information about the code area. The information available to view includes the following:

- visibility into the contents of XMLHttpRequest requests and response data
- visibility into data traffic for web-based content, including elements such as images, CSS, JavaScript, HTML, and objects
- reporting on the time to load for web-based content, including elements such as images, CSS, JavaScript, HTML, and objects

Using the BlackBerry Web Plug-in for Eclipse

The BlackBerry Smartphone Simulator includes the BlackBerry® smartphone applications that are typically available on BlackBerry smartphones and enables you to load and test your web pages in the BlackBerry Browser. From within Eclipse®, you can set break points, debug, and step through linked-in and inline JavaScript code in HTML documents. The BlackBerry Browser page in the BlackBerry Smartphone Simulator updates as you step through your code.

Start and stop debugging

1. To start debugging, on the Run menu, click Run.
2. To stop debugging, on the Run menu, click Terminate.
3. To pause debugging, on the Run menu, click Suspend.
4. To resume debugging, on the Run menu, click Resume.

View profiling data for web project resources

The Resources Loaded view displays the following information for HTML, JavaScript, and style pages, as well as images and other objects:
• full path information
• size of the resource
• load time for the resource in milliseconds

1. In Eclipse, on the Window menu, click Show View > Other.
2. Expand BlackBerry Web.
3. Click Resources Loaded.
4. Click OK.

View HTTP request and response data

The XmlHttpView view displays detailed information for HTTP requests and responses during web page debugging.

1. In Eclipse, on the Window menu, click Show View > Other.
2. Expand BlackBerry Web.
3. Click XmlHttpView.
4. Click OK.
5. In the XmlHttpView window, double-click an XML response item to see the corresponding XML request item.

Using the BlackBerry Web Plug-in for Microsoft Visual Studio

From within Microsoft Visual Studio, you can set break points, debug, and step through linked-in and inline JavaScript in ASP.NET projects. The BlackBerry Browser web page in the BlackBerry Smartphone Simulator updates as you step through your code. You can download and install multiple BlackBerry Smartphone Simulators for use with the BlackBerry Web Plug-in for Microsoft Visual Studio.

Note:
Before you debug BlackBerry web pages, you must enable JavaScript support in the BlackBerry Browser options from within the BlackBerry Smartphone Simulator.
Start and stop debugging

1. To start debugging, on the Debug menu, click Start Debugging.
2. To stop debugging, on the Debug menu, click Stop Debugging.

View profiling data for web project resources

The Web Resource Profiling window displays information for HTML, JavaScript, style page, and image resources in your web project. You can view the following information:

- full path information
- size of the resource
- load time for the resource, in milliseconds

2. Click a tab for the type of resource you want to view.
3. Click a column heading to sort the resources in ascending or descending order. You cannot sort images by thumbnail view.
4. Double-click a resource to open it in the editor.

View HTTP request and response data

The XHR Watch window displays detailed information for the HTTP requests and responses generated during web page debugging.

1. On the Debug menu, click Windows > BlackBerry XHR Watch while debugging.
1. What is the main function of the profiling tool?

2. List three types of information that you can view using the profiling tool.

3. List the profiling data available for web project resources.

4. The BlackBerry Smartphone Simulator must be running before you can use the profiling tool. True or false?

5. When you use the BlackBerry Web Plug-in for Microsoft Visual Studio to profile code, what BlackBerry Browser option must you enable in the BlackBerry Smartphone Simulator?
**Answers**

1. To identify bottlenecks in the code

2. Contents of XMLHttpRequest requests and response data
   - Data traffic for web-based content
   - Reporting on the time to load for web-based content

3. Full path information
   - Size of the resource
   - Load time for the resource, in milliseconds

4. True

5. JavaScript support
Defining queues for offline form submission

If you define form-submission queues, BlackBerry smartphone users can complete and submit forms and continue browsing without waiting for the form to be submitted or worrying about whether they are in a wireless coverage area. Users can load an HTML form (or a WML page with inputs) in the browser, fill in the values, and then submit the form to an offline queues list. The browser continuously processes any queued forms and submits the forms in the background.

If the BlackBerry smartphone is outside a wireless coverage area, users can still fill in and submit several forms (possibly for different queues). The browser queues the form requests and submits them when the smartphone is back in coverage.

After forms are submitted, user responses are stored by the browser. Users can open the queue list and click a request to view the response. The following HTTP headers allow you to create a form queue:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-rim-queue-id</td>
<td>Yes</td>
<td>Specifies the Offline Form Queue to which any GET or POST requests from form submissions on this page should go. The value may be any text string.</td>
</tr>
<tr>
<td>x-rim-next-target</td>
<td>No</td>
<td>Specifies the next page to load after sending any GET or POST requests resulting from this page to the Offline Form Queue. The value may be any valid web address.</td>
</tr>
<tr>
<td>x-rim-request-title</td>
<td>No</td>
<td>Specifies the label used to identify this request in the Queue view page. The value may be any text string. By default, the request is identified using the title of the page.</td>
</tr>
<tr>
<td>x-rim-request-id</td>
<td>No</td>
<td>Specifies whether the browser generates a unique ID and adds it as an HTTP header for every offline request resulting from this page. The value may be a boolean True or False. By default, this value is True.</td>
</tr>
<tr>
<td>x-rim-request-date</td>
<td>No</td>
<td>Specifies whether the browser generates a time stamp and adds it as an HTTP header to every offline request resulting from this page. The value may be a boolean True or False. By default, this value is True.</td>
</tr>
</tbody>
</table>

You can create form queues using these headers either by creating an HTTP property file or by adding the queuing parameters directly to the HTML or WML page.
Create an HTTP header property file

By creating an HTTP property file in which you define the queuing parameters, you can create and manage multiple form queues in a single location. However, you must properly configure your web server to send the headers when the web page containing the form is requested.

To create a queue for a form on stock-monitor.xhtml, for example, you can define the queuing parameters as follows:

```html
<Files stock-monitor.xhtml>
  Header set cache-control max-age=2592000
  Header set x-rim-queue-id Register
  Header set x-rim-request-title "Stock Monitor"
  Header set x-rim-next-target success.xhtml
</Files>
```

You can add queuing parameters for additional forms within the same header file. The following example shows a header file for the offline form queue .htaccess:

```html
<Files stock-monitor.xhtml>
  Header set cache-control max-age=2592000
  Header set x-rim-queue-id Register
  Header set x-rim-request-title "Stock Monitor"
  Header set x-rim-next-target success.xhtml
</Files>
<Files stock_monitor.wml>
  Header set cache-control max-age=2592000
  Header set x-rim-queue-id Register
  Header set x-rim-request-title "Stock Monitor"
  Header set x-rim-next-target success.wml
</Files>
<Files success.xhtml>
  Header set cache-control max-age=2592000
</Files>
<Files success.wml>
  Header set cache-control max-age=2592000
</Files>
```

Add queuing parameters directly to the web page

HTML/XHTML pages and WML pages handle queuing parameters differently.
**HTML and XHTML pages**

In HTML or XHTML, queuing parameters are added using hidden `<input>` elements:

```html
<input type="hidden" name="x-rim-queue-id" value="Register" />
<input type="hidden" name="x-rim-request-title" value="Stock Monitor" />
<input type="hidden" name="x-rim-next-target" value="success.xhtml" />
```

**WML pages**

In WML, queuing parameters are added using `<postfield>` elements:

```html
<input type="hidden" name="x-rim-queue-id" value="Register" />
<input type="hidden" name="x-rim-request-title" value="Stock Monitor" />
<input type="hidden" name="x-rim-next-target" value="success.wml" />
```
1. List two benefits of defining queues for offline form submission.

2. Which HTTP headers are required when you create a form queue? Choose all that apply.
   A. x-rim-queue-id
   B. x-rim-next-target
   C. x-rim-request-title
   D. x-rim-request-id
   E. x-rim-request-date

3. List two ways to create a form queue using HTTP headers.

4. Which element do you use to add queueing parameters to a WML page?
Answers

1. Users can continue browsing without waiting for a submission; users can submit the form when there is no wireless coverage.

2. A

3. Create an HTTP property file; add directly to an HTML or WML page.

4. <postfield>
Optimization techniques used by network gateways

The BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing network gateways are specifically designed to deliver content to the BlackBerry Browser. To help maximize efficiency and minimize the bandwidth requirements of content that is sent over the wireless network, these network gateways use optimization techniques unique to the BlackBerry Infrastructure.

The BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing network gateways optimize content in the following ways:

- by reading the existing content in the BlackBerry Browser cache to determine if requested content is already stored on the device
- by preprocessing requested content and filtering out unsupported or unnecessary markup or resources
- by converting images into an appropriate format for display on the BlackBerry smartphone, reducing the color depth of images, and scaling images for display on a small screen size by compressing content for efficient delivery over the wireless network

Some WAP network gateways might optimize content and might convert HTML to XHTML or limit the size of images. WAP network gateways are not designed for a particular user agent and therefore cannot offer the same level of optimization for the BlackBerry Browser as the BlackBerry MDS Connection Service or the BlackBerry Internet Service Browsing network gateways. Content delivery times are typically slower over WAP network gateways. WAP network gateways do not remove unsupported elements before they send the content. As a result, the BlackBerry Browser must parse more content on the BlackBerry smartphone, which increases the time it takes to render the content.

Content transcoding

The BlackBerry MDS Connection Service and BlackBerry Internet Service Browsing network gateways transcode data into formats that are more suitable for sending over the wireless network and for rendering in the BlackBerry Browser.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>markup languages</td>
<td>Markup languages are transcoded in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• HTML and XHTML are simplified and transcoded into a format similar to WBXML.</td>
</tr>
<tr>
<td></td>
<td>• WML is transcoded into WMLC.</td>
</tr>
<tr>
<td></td>
<td>• SVG is transcoded into PME, a proprietary binary file format that BlackBerry smartphones can render.</td>
</tr>
</tbody>
</table>
Content preprocessing and filtering

The BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing network gateways can check the content type that a BlackBerry smartphone supports based on the version of the BlackBerry Device Software that the smartphone is running. The network gateway preprocesses the HTML, removes any unsupported elements or attributes, and sends the HTML to the BlackBerry smartphone as it is processed; it does not wait until all the HTML is completely processed. The network gateway retrieves any images from the content server while it preprocesses the HTML.

The network gateway reads the browser configuration settings on the BlackBerry smartphone to check if the user has turned off support for markup, such as tables, or external resources, such as JavaScript, style sheets, or embedded media. If support for a resource is turned off, the network gateway does not retrieve the resource and does not send it over the wireless network to the smartphone, to minimize the use of bandwidth.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scripts</td>
<td>JavaScript (internal or external) is transcoded into JavaScript bytecode.</td>
</tr>
<tr>
<td></td>
<td>WMLScript is transcoded into compiled WMLScript.</td>
</tr>
<tr>
<td>images</td>
<td>All .gif files are transcoded into PNG file format.</td>
</tr>
<tr>
<td></td>
<td>If the BlackBerry smartphone does not support native JPG files, JPG files are converted to PNG file format.</td>
</tr>
<tr>
<td>Java applications</td>
<td>JAR files are converted into COD file format.</td>
</tr>
</tbody>
</table>
Image optimization

The BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing network gateways check the physical characteristics of the destination BlackBerry smartphone, and scale the image and reduce the color depth as appropriate.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| image size           | To reduce the use of bandwidth and improve display, the network gateway scales images. When the network gateway delivers content for display in the Page view, the network gateway scales images proportionally with the content. When the network gateway delivers content for display in the Column view, the network gateway scales images in the following ways:  
  - Horizontal scaling: Images are scaled to fit the width of the BlackBerry smartphone screen, less 5 pixels for the scroll bar.  
  - Vertical scaling: Images are scaled to a maximum height of twice the screen height.  
  The network gateway maintains the aspect ratio when it scales images. Users can download the full image. The BlackBerry Browser sends a secondary request to the network gateway to deliver the image in its original form, with no optimization. |
| image color depth    | The BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing network gateways reduce the color depth of the image to match the number of colors that the destination BlackBerry smartphone supports. This results in smaller file sizes and minimizes the use of bandwidth.                                                                                           |
BlackBerry smartphones running earlier versions of the BlackBerry smartphone software, the network gateway sets the image quality as medium.

**Progressive downloading of images**

The BlackBerry MDS Connection Service network gateway (version 4.1.3 or later) and the BlackBerry Internet Service Browsing network gateway support progressive downloading of images. When the network gateway compresses an image, it also divides the file into segments. The network gateway sends the file segments to the BlackBerry Browser incrementally.

As the web page is rendered in the BlackBerry Browser, low quality placeholder versions of the images are rendered quickly. After the entire page is rendered, the browser receives the additional image file segments. As the additional file segments arrive, the placeholder images are enhanced until the images meet the image quality that is specified by the user.
1. WAP network gateways remove unsupported elements before sending content. True or false?

2. What format does the BlackBerry MDS use to send Java applications over the wireless network?

3. List two ways that the gateway can optimize images.
Answers

1. False

2. COD

3. Decrease image size; decrease color depth.
Many factors influence how content is rendered on the BlackBerry smartphone. These factors include the network gateway, the browser configuration, and amount of memory in the wireless device, and the color depth of images. In addition, wireless devices have smaller screens and slower processors than desktop computers, which affect how web content is rendered.

By following basic design principles—such as organizing your web content, choosing the most appropriate markup language, creating effective images, and minimizing the need for backward navigation— you can create efficient web content that provides a user-friendly experience for users of mobile devices.

In addition to these design principles, you can optimize the performance of your web site by using the BlackBerry IDE profiler tool to identify and address bottlenecks, and by defining queues for offline form submission.

Optimization techniques are also supported by the network gateway. The BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing network gateways are specifically designed to deliver content to the BlackBerry Browser.

The BlackBerry MDS Connection Service and the BlackBerry Internet Service Browsing network gateways optimize content in the following ways:

- by reading the existing content in the BlackBerry Browser cache to determine if requested content is already stored on the device
- by preprocessing requested content and filtering out unsupported or unnecessary markup or resources
- by converting images into an appropriate format for display on the BlackBerry smartphone, reducing the color depth of images, and scaling images for display on a small screen size by compressing content for efficient delivery over the wireless network
1. List the three markup languages that you can use to develop web content for a mobile device, and describe the advantages and disadvantages of each.

2. List and describe the HTTP headers that you can use to create a form queue.

3. List and describe three ways that network gateways can optimize the delivery of web content.

4. Describe how the BlackBerry MDS uses browser configuration settings to optimize the delivery of web content.

5. Describe the steps in the process of progressively downloading images.