

**ORACLE®**

---

**ENDECA**

Oracle Endeca for Mobile  
Getting Started Guide – Web Application

# Copyright and Disclaimer

Copyright © 2003, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Rosette® Linguistics Platform Copyright © 2000-2011 Basis Technology Corp. All rights reserved.

Teragram Language Identification Software Copyright © 1997-2005 Teragram Corporation. All rights reserved.

## Getting Started Guide – Web Application

The Oracle Endeca for Mobile Web Application sits on top of the Mobile API server. It is recommended to read the “Getting Started Guide – Mobile API” before reading this one.

The Oracle Endeca for Mobile Web Application leverages all the underlying controllers and models provided by the Endeca Mobile API. To make the application complete, the endeca-mobile-web-application.zip package contains all the view tier JSP, CSS, JavaScript, and image files to support a mobile web application. In addition, the package contains a Java library to perform device detection and URL redirection. This package is used in the mobile web to determine device info and adapt the application to different profiles. The three profiles used are “simpleHTML”, which includes all phones that don’t fully support a full HTML/CSS and JavaScript standard, “fullHTML”, and “HTML5”. The device detection jar can also be used on the main desktop web application to determine if the user is accessing the desktop url from a mobile phone, and to retrieve the mobile web url for the desktop url (then either redirect, or fetch the mobile content and return it to the user)

The determination of device profile is used throughout the JSP code to optionally include CSS and JavaScript, or to switch between simple HTML and more advanced HTML for the same functionality. The code has been developed, however, to reduce the use of these flags as much as possible, in favor of writing HTML that is fairly simple and most likely to render well on both simple and more advanced mobile browsers.

## Package Structure

Below is the package structure of the Oracle Endeca for Mobile Web Application:

```
/views_m
  /components
    / shared
      /css
      /images
      /javascript
      html_header.jspf
      menu_header.jsp
      ...
    /MobileCategoryPage
      /component.jsp
      ...
  search_main.jsp
  search_detail.jsp
  cart_list.jsp
  checkout.jsp
  brands.jsp
  user_login_form.jsp
  more_menu.jsp
  ...
/lib
  mobile-device-detection-[version].jar
```

## Request Scoped Objects/Attributes

Request Key	Class	Description
pageResult	com.endecia.b2creference.content.PageResult	This object contains a lot of key

		objects, such as the ENEQuery, ENEQueryResults, UriState, UriConfig, UriFormatter, QueryResultsComponent, and RecordComponent
uiclQueryResults	com.endeca.soleng.uiComponentLibrary.web.urlFormatter.components.QueryResultsComponentUrl	This is included in for convenience, since most of the views are pulling their Endeca data from UICL objects
urlState	com.endeca.soleng.urlformatter.UriState	This is also included for convenience, since the object is also stored in the PageResult
deviceInfo	com.endeca.mobile.services.detection.DeviceInfo	This provides access to device info, such as profile ('simpleHTML', 'fullHTML', 'HTML5'), and OS.
contentItem	com.endeca.content.ContentItem	This is used by the Experience Manager powered components

## Required Configuration Changes

### search\_main.jsp

This is the main page used by the SearchController, which handles search, guided navigation, and Experience Manager powered pages.

You will need to change the sort options in this file:

```
<option value="product.price|1" ${param.Ns eq 'product.price|1' ? 'selected' : "" }>Price</option>
<option value="product.review.avg_rating|1" ${param.Ns eq 'product.review.avg_rating|1' ?
'selected="selected" : "" }>Rating</option>
```

### search\_records.jspf

This page fragment renders the records list, and the Endeca properties need to be updated for your dataset:

```
...
<div class="productBrand">
    ${rec.localizedProps['product.brand.name']}
</div>
...
```

### search\_detail.jsp

This page renders the record detail page, and the Endeca properties need to be updated for your dataset:

```
...
<div class="productTitle">
    ${rec.props['product.name']}
</div>
...
```

### components/ shared/search\_header.jspf

If you are not using the SearchInterfaceQueryProcessor in the Endeca Mobile API, you will need to set the search interface manually in a hidden input field:

```
<input type="hidden" name="Ntk" value="All" />
```

You may also need to update the BaseController.java class and set the search interface there:

```
searchUrlState.setParam("Ntk", "All");
```

### **more\_menu.jspf and components/shared/menu\_header.jspf**

The main menu is split across these two files. For devices that support the 'fullHTML' profile, the more menu will be included in a hidden section on the page and made visible through JavaScript when clicking on the more link. For older class devices, the more link will go to a separate page.

### **components/shared/images/icon.png**

For iPhone, this image is used as the image if a user bookmarks the page and adds it to the Home Screen. If you want to rename this image, or remove the functionality, it is referenced in

### **components/shared/html\_header.jspf:**

```
<link rel="apple-touch-icon" href="{STATIC_RESOURCE_PATH}/images/icon.png" />
```

You can also change the configuration here if you don't want the icon shine using:

```
<link rel="apple-touch-icon-precomposed" href="{STATIC_RESOURCE_PATH}/images/icon.png" />
```

## **Compressing CSS and JavaScript for Production**

To increase performance of the mobile web application, the Yahoo CSS and JavaScript compressor is included with the endeca-mobile-web-application.zip package, along with a set of scripts that automate the compression of JavaScript and CSS files in **/WEB-INF/views\_m/components/shared/javascript** and **/WEB-INF/views\_m/components/shared/javascript/css**. To run these scripts:

1. Make sure **yuicompressor/script.properties** has been setup to point to your mobile\_shared directory.
2. Run **setup\_for\_production** – this will move the .js and .css files from javascript and css directories to new javascript\_full and css\_full directories.
3. Run **compress** – this will compress the .js and .css files in the javascript\_full and css\_full directories, putting the compressed versions back in the javascript and css directories.

For convenience, a **setup\_for\_development** script is included that will copy the uncompressed .js and .css files from javascript\_full and css\_full back into the javascript and css folders, and remove the javascript\_full and css\_full directories.

## **Mobile CSS**

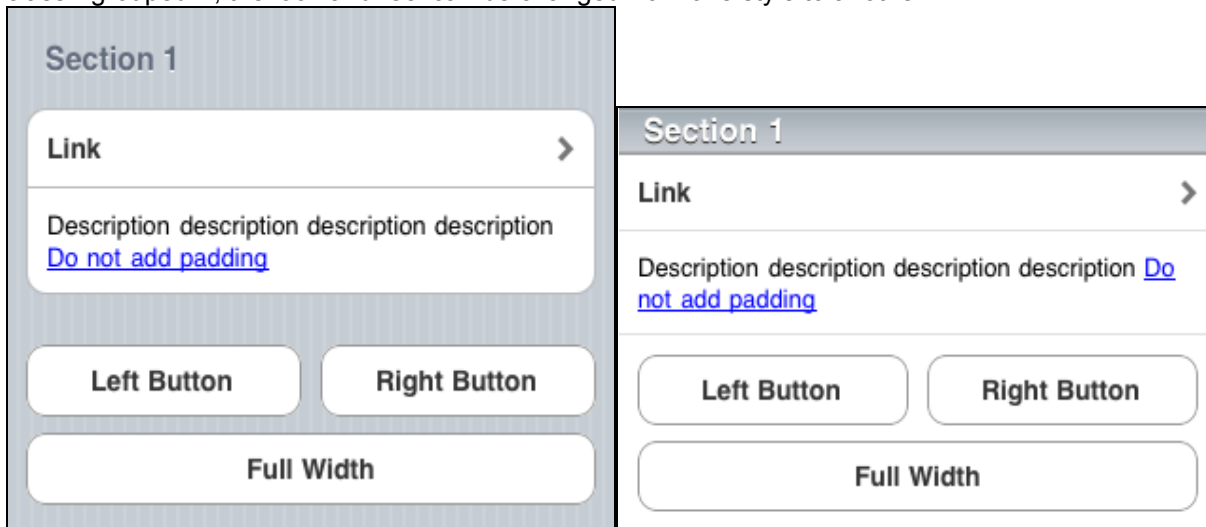
```
/views_m/components/shared/css  
  cart.css  
  categories_full.css
```

categories.css  
 checkout.css  
 detail.css  
 forms.css  
 global.css  
 navigation.css  
 records\_full.css  
 records.css  
 search.css  
 stores\_simple.css  
 stores.css  
 table\_grouped.css  
 table\_plain.css  
 table.css

Some of these CSS files are specific to certain views: cart.css, categories.css, checkout.css, detail.css, navigation.css, and records.css. The files with ‘\_full.css’ at the end are only included when the device profile is not ‘simpleHTML’.

The forms.css, and table\_grouped.css, table\_plain.css, and table.css provide a general set of CSS styles for forms and tables (not HTML tables, but styles that match the iPhone look and feel for Plain and Grouped Table Views).

Below are a two screenshots, and the corresponding HTML showing the different components that are available through the table CSS classes. By simply wrapping the contents in <div class="plain"> or <div class="grouped">, the look and feel can be changed from one style to another.



```
<div class="plain">
  <div class="sectionTitle">Section 1</div>
  <div class="section">
    <div class="sectionCell disclosure first">
      <a href="">Link</a>
    </div>
    <div class="sectionCell last">
      <div class="contents">
        Description description description description
        <a class="webLink">Do not add padding</a>
      </div>
    </div>
  </div>
```

```

</div>
<div class="sectionSpacer"> </div>
<div class="buttonRow">
  <div class="left"><div class="button"><a href="">Left Button</a></div></div>
  <div class="right"><div class="button"><a href="">Right Button</a></div></div>
</div>
<div class="button">
  <a href="">Full Width</a>
</div>
<div class="sectionSpacer"> </div>
</div>

```

## Mobile HTML/CSS Best Practices

You can read up on some articles about mobile web development. Both of these are a bit dated with the wave of new modern mobile devices like the iPhone, Android, the latest BlackBerry's and the Palm Pre/Pixi, but they are still worth a read for the context.

<http://www.passani.it/gap/>

[http://wiki.forum.nokia.com/index.php/Device\\_and\\_feature\\_detection\\_on\\_the\\_mobile\\_web](http://wiki.forum.nokia.com/index.php/Device_and_feature_detection_on_the_mobile_web)

For mobile testing, you can use a User-Agent switcher plugin for Firefox to mimic different devices.

<http://chrispederick.com/work/useragentswitcher/>

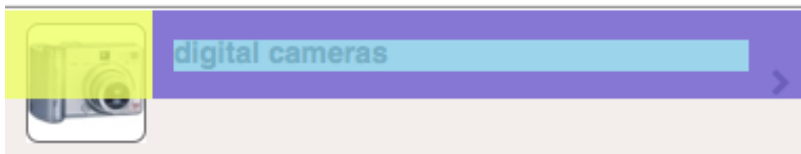
You can lookup User-Agent strings on Wikipedia:

[http://en.wikipedia.org/wiki/List\\_of\\_user\\_agents\\_for\\_mobile\\_phones](http://en.wikipedia.org/wiki/List_of_user_agents_for_mobile_phones)

Let's take a look at a few HTML/CSS best practices that we've come up with through research and testing.

### Use float and left-margin together to lay out a cell:

The layout below can be accomplished in a number of ways on a desktop browser (both .iconURL and .text floated left, .text with a 80px left-padding, etc). To achieve maximum browser compatibility, however, it's best to lay this out with a float and left-margin:



```

<div style="" class="sectionCell even disclosure">
  <div style="width: 57px; height: 57px; margin-top: 6px; margin-bottom: 6px;" class="iconURL">
    <img width="57" height="57" style="left: 0px; top: 0px;">
    <div class="overlay"></div>
  </div>
  <div style="margin-left: 70px;" class="text">
    <a href="#" class="noLink">digital cameras</a>
  </div>
  <div style="clear: both;"></div>
</div>

```

```

.iconURL {
    float:left;
    height:57px;
    margin:10px;
    width:57px;
}

.sectionCell .text {
    padding:14px 30px 13px 10px;
}

```

**Use background-image CSS properties for images that aren't necessary for older browsers:**

Another good practice is to put icons like the disclosure icon shown in the above example, or icons like checkmarks or plus/add icons into the CSS as background-image properties. These will get ignored by older mobile browsers, and don't add extra complexity to your HTML.

```

div.disclosure {
    background-image:url("../images/listArrow.png");
    background-position:right center;
    background-repeat:no-repeat;
}

```

**Do not use block tags for anything you want to retain a horizontal layout on older browsers, and something that floats right should always be last:**

In web browsers, there are a number of ways to achieve the screenshot below, and it doesn't really matter where you put the .RefinementCount element, as long as it's floating right. It also doesn't matter if you use a div or a span, and the <a> tag can have a "display:block" style without a problem.

Older mobile browsers, on the other hand, often don't support the float or display CSS attributes. Therefore, if you want to maintain the order in the screenshot below (checkbox, refinement label, refinement count), you need to make sure that if those CSS attributes are not applied that the html tags used and the order of those tags will result in the elements naturally lining up inline in the same order.



```

<div class="contents">
    <input type="checkbox" class="RefinementCheckbox">
    <a href="#" class="noLink Refinement">3</a>
    <span class="RefinementCount">&nbsp;&nbsp;&nbsp;(3)</span>
    <div style="clear: both;"></div>
</div>

```

```

input.RefinementCheckbox {
    float:left;
    margin-right:8px;
}

```



```
div.sectionCell a.Refinement {
    display:inline;
    float:left;
    margin-top:5px;
}
```

```
span.RefinementCount {
    float:right;
    margin-top:5px;
}
```

## Mobile JavaScript Best Practices

In general, you should avoid the use of JavaScript. JavaScript required for the 'fullHTML' device profile should only be included if the device is that profile, or else it may break all the other JavaScript that would normally work on an older mobile browser.

### Don't use JavaScript to submit a form:

Forms should always use the action for maximum compatibility

```
<form action="/mobile/search/_/N-/?Ntk=All" name="search_form">
    <input type="text" data-defaultvalue="Search..." class="SearchTerms" maxlength="100"
name="Ntt">
    <input type="submit" value="Go" class="SearchSubmit">
    <input type="hidden" value="search" name="tab">
    <input type="hidden" value="All" name="Ntk">
</form>
```