



# BEA WebLogic Portal™

## MobileAware Interaction Server Administration Guide

# Copyright

Copyright © 2004-2005 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, BEA WebLogic Server, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic JRockit, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

# Contents

## MobileAware Interaction Server Administration Guide

About This Manual .....	1
Further Reading .....	1
System Administration.....	2
Deploying an Application.....	2
Creating the Deployment Site .....	2
Configuring the Web Deployment Descriptor.....	3
Configuring the MobileAware Interaction Server .....	6
Configuring the MobileAware Interaction Server Filter .....	6
The mis.properties File .....	6
Configuring the MobileAware Device Database Connection .....	7
Configuring URL Compression.....	12
Error Handling (configuration optional) .....	14
Configuration Mode (configuration optional).....	15
Generated URLs .....	15
Strict Attribute Handling and Delivery (configuration optional) .....	17
Diagnostics Subscriptions (configuration optional) .....	17
Back to Top (configuration optional) .....	19
Content Length Settings .....	19
Disallowed Output Encodings (configuration optional) .....	20
Optimizing Performance with the JSP Tag Library (configuration optional) ..	20

BEA WebLogic Portal Settings (configuration mandatory for BEA Portal deployment) . . . . .	23
The ContentAssembly.properties File . . . . .	24
The oscache.properties File . . . . .	27
Configuring the Error Pages . . . . .	28
Resource Administration . . . . .	29
Getting Started . . . . .	29
Quick Start. . . . .	29
Logging In. . . . .	30
Using the System Monitor. . . . .	30
Using the Administration Console Toolbar. . . . .	30
Managing Devices . . . . .	31
About Device Profiles . . . . .	31
CC/PP Attributes . . . . .	32
MobileAware Attributes . . . . .	33
About the Device Database . . . . .	33
The Organization of the Device Profiles. . . . .	33
Device Recognition . . . . .	35
Comparing Incoming Requests to Device Attributes . . . . .	35
Multiple Header Strings . . . . .	36
No Match. . . . .	36
Transforming Content. . . . .	36
Tailoring Content . . . . .	37
Creating and Modifying Device Profiles . . . . .	37
Creating Device Profiles . . . . .	38
Adding a Device . . . . .	38
Adding a Device Attribute . . . . .	38
Configuring an Attribute . . . . .	39

Modifying a Device Profile . . . . .	39
Viewing an Attribute . . . . .	39
Deleting Devices . . . . .	39

## Appendix A – Device Attributes

The CC/PP Attributes . . . . .	41
The MobileAware Attributes. . . . .	56
Deprecated Attributes . . . . .	66

## Appendix B – web.xml Sample File

# MobileAware Interaction Server Administration Guide

## About This Manual

This manual introduces you to the various tasks required to configure and manage the MobileAware Interaction Server. The manual is divided into three parts:

### **System Administration**

Explains how to set up and configure the MobileAware Interaction Server. It describes the various files used by the MobileAware Interaction Server and the changes you need to make during configuration.

### **Resource Administration**

Explains how to set up device profiles and how to manage the various device profiles. These profiles are stored in the device database and are managed through the Administration Console, a Java-based GUI that provides a convenient way of setting up, retrieving and modifying the various attributes associated with each profile.

### **Appendix A – Device Attributes**

### **Appendix B – web.xml Sample File**

## Further Reading

For further information on the MobileAware Interaction Server, see the following manuals:

- MobileAware Interaction Server, BEA WebLogic Edition Installation Guide
- MobileAware Interaction Server, BEA WebLogic Edition Getting Started Tutorials

- MobileAware Interaction Server, BEA WebLogic Edition User Guide
- BEA Sample Workshop Mobility Project Guide
- BEA Sample Mobility Portal Guide

## System Administration

### Deploying an Application

In order to develop Java Server Pages or Java Servlets, regardless of whether they are mobilised or not, you must establish a server-side environment that interprets, compiles, and executes the pages that you write. This section describes how to deploy a mobilised web application with the MobileAware Interaction Server servlet filter.

There are two key steps to deploying the MobileAware Interaction Server applications in a servlet environment:

- [Creating the Deployment Site](#)
- [Configuring the Web Deployment Descriptor](#)

### Creating the Deployment Site

Servlet containers such as BEA Tomcat and WebLogic support the Java Servlet 2.3 specification that standardises how the various components in a web application are organised. Some things to remember:

- You store all the files for a web application under a root directory. This root directory also serves as the document root for this web application. Usually, you would place the root directory within the applications directory of your container. For example:  
..\jakarta-tomcat\webapps\your\_webapp
- You store all application files (such as the JSP pages, HTML pages and images) under the root directory.
- You must create a special directory named "WEB-INF" under the root directory that contains the information describing the web application and how it should be configured and how it should behave.
- For a new MobileAware Interaction Server deployment, copy the contents of <installdir>\lib directory to WEB-INF\lib.

**Note:** If you developed your mobilized application in BEA WebLogic Workshop with MobileAware Interaction Server, BEA WebLogic Edition installed, “Enable Multi-Channel” has already copied any necessary MobileAware Interaction Server files into the WEB-INF directory.

The following table shows the contents of the WEB-INF directory:

**Table 1 WEB-INF Directory Contents**

WEB-INF Contents	Description
/WEB-INF/web.xml	The deployment descriptor for the application.
/WEB-INF/classes/*	A directory for Java utility classes and other files such as the mis.properties and oscache.properties files.  See the next section, “”, for further details on how to set up this directory.
/WEB-INF/lib/*.jar	A directory for Java ARchive (JAR) files that contain servlets, JavaBeans, and other utility classes for the web application. The container automatically loads these classes. Copy the contents of the <installdir>\lib to this folder.

For additional information on deploying applications on BEA WebLogic Platform, Please refer to Developing Web Applications for BEA WebLogic Server documentation: <http://e-docs.bea.com/wls/docs81/webapp/index.html>:

## Configuring the Web Deployment Descriptor

Before deploying a web application, you need to place a deployment descriptor (web.xml) into the WEB-INF directory. This file pulls together all the components of the Web application.

The following table describes the key elements in the web.xml file:

**Table 2 Key Elements in web.xml File**

Element	Description
<web-app>	This is the root element for the deployment descriptor.
<display-name>	Specifies the name to be displayed for the application.

<code>&lt;filter&gt;</code>	Names the filter used by this web application and the parameters it receives.
<code>&lt;filter-mapping&gt;</code>	Specifies which URL pattern is mapped to the servlet.

### Configuring the MIS Filter and Filter Mapping

An example of a `web.xml` file that has been configured for the MobileAware Interaction Server filter is shown below followed by an explanation of the settings.

```

<filter>
    <filter-name>mobilityFilter</filter-name>
    <display-name>MIS Mobility Filter</display-name>
    <description>MIS Mobility Filter</description>
    <filter-class>com.mobileaware.mcp.MobilityFilter</filter-class>
    <init-param>
        <param-name>propertiesname</param-name>
        <param-value>/mis.properties</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>mobilityFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

**Table 3 MobileAware Interaction Server web.xml Filter Settings**

Setting	Description
<code>&lt;filter-name&gt;</code>	The name of the filter used by this web application: mobilityFilter
<code>&lt;display-name&gt;</code>	The name of the filter to be displayed: MIS Mobility Filter

<description>	A description of the filter: MIS Mobility Filter
<filter-class>	The class file containing the filter: com.mobileaware.mcp.MobilityFilter
<param-name>	A name for each parameter to be passed to the MobileAware Interaction Server: propertiesname
<param-value>	A value for each corresponding parameter name. In this case, the name and location of the properties file. /mis.properties

**Table 4 MobileAware Interaction Server web.xml Filter Mapping Settings**

Setting	Description
<filter-name>	Specifies which URL pattern is to be mapped for the servlet specified by <filter-name>: mobilityFilter
<url-pattern>	The URL pattern to follow: /*

For more details on the Java Server specification and how to configure the web.xml file, please refer to the following URLs:

- Java Servlet Specification:  
<http://www.jcp.org/aboutJava/communityprocess/final/jsr053/>
- General Java servlet technology documents:  
<http://java.sun.com/products/servlet/docs.html>

Developing Web Applications for BEA WebLogic Server documentation:  
<http://e-docs.bea.com/wls/docs81/webapp/index.html>

The MobileAware Interaction Server provides a set of sample JSP error handling pages and images:

Copy the error pages to your web application directory; you can place them in their own directory below the webapp root if you prefer.

Modify the following lines in the properties file so that they point to the appropriate error handlers.

```
<filter-name>mobilityFilter</filter-name>  
    <url-pattern>/*</url-pattern>
```

## Configuring the MobileAware Interaction Server

### Configuring the MobileAware Interaction Server Filter

Configuring a MobileAware Interaction Server filter for each web application involves creating an `mis.properties` file that specifies:

- the database that the MobileAware Interaction Server needs to connect with
- the error pages to be returned in the event of any HTTP- or WAP-generated errors
- number of additional configuration settings as described in “Configuring the MobileAware Interaction Server”

There is a sample properties file, `mis.properties.sample`, in the `lib` folder of the installation directory. Copy this file into `WEB-INF/classes` folder and make the necessary changes for configuring the database and the error pages.

You can rename the file to a name of your own choosing, but you need to specify the name in `<param-value>` section of the `web.xml` file that has been configured for the filter. See the section “Configuring the MIS Filter and Filter Mapping” for further details.

This section introduces the various properties used to manage the behaviour of the MobileAware Interaction Server. As an administrator, you will need to configure some of these properties to adjust the behaviour of the MobileAware Interaction Server.

Some of these properties will have been set during the install process, while others can be configured later when you want to activate other new features.

### The `mis.properties` File

The main configuration file used by the MobileAware Interaction Server, which contains a number of user-modifiable parameters, is the `mis.properties` file.

This file can be found in the WEB-INF/classes folder. It is a plain text file that can be edited in any text editor.

This section describes configuration settings for the `mis.properties` file.

## Configuring the MobileAware Device Database Connection

The following device database properties must be configured in order for that the MobileAware Interaction Server to can successfully communicate with the MobileAware Device Database used:

**Table 5 Device Database Properties Settings**

Property	Description
<code>deviceDB.driver</code>	<p>Location of the JDBC driver to be used by the MobileAware Interaction Server to gain access to the database.</p> <p>This property also has the effect of informing the MobileAware Interaction Server which database it is connected to.</p> <ul style="list-style-type: none"> <li>• For Oracle, set to: <code>oracle.jdbc.driver.OracleDriver</code></li> <li>• For MySQL, set to: <code>org.gjt.mm.mysql.Driver</code></li> <li>• For Postgres, set to: <code>org.postgresql.Driver</code></li> <li>• For PointBase, set to: <code>com.pointbase.jdbc.jdbcUniversalDriver</code></li> <li>• For SQL Server (with MIS deployed on BEA WebLogic only), set to: <code>weblogic.jdbc.sqlserver.SQLServerDriver</code></li> </ul> <p>To configure MIS to use the BEA WebLogic database connection pool: <code>weblogic.jdbc.pool.Driver</code></p> <p>Example: <code>deviceDB.driver:oracle.jdbc.driver.OracleDriver</code></p>

---

<code>deviceDB.url</code>	<p>The URL to use to access the Device Database.</p> <ul style="list-style-type: none"> <li>• For Oracle, set to:  <code>jdbc:oracle:thin:@&lt;oracle_host&gt;:&lt;oracle_port&gt;:&lt;oracle_database_name&gt;</code></li> <li>• For MySQL, set to:  <code>jdbc:mysql://&lt;mysql-server-ip:port&gt;/&lt;db-name&gt;?user=&lt;connect-user&gt;&amp;password=&lt;connect-password&gt;</code></li> <li>• For Postgres, set to:  <code>jdbc:postgresql://&lt;postgres_machine&gt;:&lt;postgres_port&gt;/&lt;postgres_database_name&gt;</code></li> <li>• For PointBase, set to:  <code>jdbc:pointbase:server://&lt;pointbase_machine&gt;:&lt;pointbase_port&gt;/cajun</code></li> <li>• For SQLServer (with MIS deployed on BEA WebLogic only), set to:  <code>jdbc:bea:sqlserver://&lt;sqlserver_host&gt;:&lt;sqlserver_port&gt;;databaseName=&lt;sqlserver_database_name&gt;</code></li> <li>• When using WebLogic database connection pool, set to:  <code>jdbc:weblogic:pool:&lt;poolname&gt;</code>                      Example: <code>deviceDB.url:</code>  <code>jdbc:oracle:thin:@oracle_host:1521:mySID</code></li> </ul>
<code>deviceDB.user</code>	<p>Username used by the MobileAware Interaction Server to access the database server.</p> <p><b>Note:</b> For MySQL, this property is left blank.</p> <p>Example: <code>deviceDB.user: user</code></p>
<code>deviceDB.password</code>	<p>Password of user used by the MobileAware Interaction Server to access the database server. For MySQL, this property is left blank.</p> <p>Example: <code>deviceDB.password: password</code></p>
<code>deviceDB.maxDBConnections</code>	<p>A numeric value indicating the number of concurrent database connections in the database pool. This is used to control the number of concurrent database connections and licenses required by the MobileAware Interaction Server. Defaults to 10. See “About Connection Pools” below for more information.</p> <p>Example: <code>deviceDB.maxDBConnections: 10</code></p>

---

---

<code>deviceDB.waitTime</code>	A numeric value indicating (in milliseconds) the time to wait for a connection from the database pool. Defaults to 5000.  Example: <code>deviceDB.waitTime: 5000</code>
<code>deviceDB.increment</code>	A numeric value indicating the number of connections to add to the pool if there are no connections currently available. If the maximum number of connections in the pool has been reached then no new connections will be added to the pool. Defaults to 1  Example: <code>deviceDB.increment: 1</code>

---

## About Connection Pools

A dynamic web site often generates HTML pages from information stored in a database. Each request for a page results in a database access. Connecting to a database is time consuming since the database must allocate communication and memory resources as well as authenticates the user and set up the corresponding security context. Setting up the individual connections can become a bottleneck.

Establishing the connection once and using the same connection for subsequent requests can therefore dramatically improve the performance of a database driven web application. Connection pooling is a technique used to avoid the overhead of making a new database connection every time an application or server object requires access to a database. Rather than making and breaking connections as required, a "pool" of database connections is maintained by the system on the server. When the MobileAware Interaction Server needs a database connection, it simply requests an available one from the pool - if none is available, a new one is created & added to the pool.

The connection pool not only grows to specified limits, but also contracts as required, closing connections that have not been used for a specified time. This avoids taking up system resources by simply holding connections that are not currently required. This also handles databases which "time-out" their connections, and prevents handing a "stale" connection to an application object.

## Configuration of XML File-based MobileAware Device Database

To configure MIS to use an XML File-based MobileAware Device Database instead of connecting to an external database (e.g. Oracle, MySQL) where the MobileAware Device Database has been installed, the database settings must be defined properly in the `mis.properties` file associated with the web application.

1. Locate the `mis.properties` file for your web application (e.g. for the sample Mobility Portal application, it is located at:  
<bea>\weblogic81\mobileaware\samples\BEAWorkshop\maportal\WEB-INF\classes\ and open it in a text editor.

2. Look for the Device Repository Type setting in the `mis.properties` file, similar to below:

```
#####  
#  
# Device Repository Type  
# -----  
# This setting indicates whether the MobileAware Device Database is  
# deployed as an XML file or installed into a JDBC database.  
# Possible values are: xml and db  
#  
# If not specified, db is assumed.  
#deviceRepositoryType: xml
```

3. Uncomment the last line so that it now reads:

```
deviceRepositoryType: xml
```

4. Look for the XML Device Repository File Location setting in the `mis.properties` file, similar to below:

```
#####  
#  
# XML Device Repository File Location  
# -----  
# This setting indicates the location of the Device Repository XML file  
# This setting must be set to the location of the xml file or the  
# classpath will be checked (see deviceXML.resourceName)
```

```
#
# Example:
#C:\\\bea81sp3\\weblogic81\\mobileaware\\database\\DeviceRepository.xml
#
#deviceXML.location:
C:\\bea81sp3\\weblogic81\\mobileaware\\database\\DeviceRepository.xml
```

5. Uncomment the 'deviceXML.location:' line and change the indicated location to the actual location of the DeviceRepository.xml file. The DeviceRepository.xml file included with MobileAware Interaction Server is located at:

<bea>\weblogic81\mobileaware\database\DeviceRepository.xml. For example,

```
deviceXML.location:
C:\\bea81sp3\\weblogic81\\mobileaware\\database\\DeviceRepository.xml
```

6. Look for the XML Device Repository Resource Name setting in the mis.properties file, similar to below

```
# XML Device Repository Resource Name
# -----
# This setting indicates the name of the Device Repository XML file that
# should be used in the classpath.
#
# The default is "/DeviceRepository.xml"
#
# Example: deviceXML.resourceName: /DeviceRepository.xml
deviceXML.resourceName: /DeviceRepository.xml
```

7. If the XML file to be used is not named DeviceRepository.xml, uncomment the 'deviceXML.resourceName:' line and change the indicated resource name to reflect the actual name of the XML file to be used. By default, DeviceRepository.xml is assumed.

8. Save the `mis.properties` file.
9. In production environment, you must now use the BEA WebLogic Administration Console to redeploy your web application. In a development environment, the web application can simply be redeployed directly from within BEA WebLogic Workshop.

### Configuring Session Encoding of URLs (configuration optional)

Where session cookies are not supported by devices or gateways, session information can be automatically encoded into URLs using the settings in the table below.

**Table 6 Session Encoding of URLs Properties Settings**

Property	Description
<code>generatedLinks.encodeSessionId</code>	<p>For URLs generated by the MobileAware Interaction Server, this property defines whether the MobileAware Interaction Server calls the application server's <code>encodeURL()</code> method to automatically append a session ID. By default, this is set to true. When this property is set to false it will stop the MobileAware Interaction Server from inserting session IDs in any content it generates. If set to false, all devices or gateways which connect to the MobileAware Interaction Server must support session cookies.</p> <p>Example: <code>generatedLinks.encodeSessionId: false</code></p>
<code>rewriteAllUrls</code>	<p>This property defines whether the MobileAware Interaction Server calls the application server's <code>encodeURL()</code> method to automatically append a session ID for URLs not generated by the MobileAware Interaction Server. By default, this is set to false. If set to false, either all such URLs must be manually encoded with the session ID or all devices or gateways which connect to the MobileAware Interaction Server must support session cookies.</p> <p>Example: <code>rewriteAllUrls: true</code></p>

### Configuring URL Compression

The URLs generated by portal frameworks and other content servers are often very long. If URL rewriting is used instead of cookies for session management the length of these URLs is extended further. Because the length of these URLs takes up valuable space within the limited memory of a small device, the output visible to the user is often very limited. In extreme cases, pages are limited to just 2 or 3 links.

To mitigate this, the MobileAware Interaction Server supports URL compression, which reduces the length of these URLs to a minimum, thereby allowing much more content to be delivered to the device. This is especially relevant where the device has limited memory but could also be important where limited bandwidth is an issue.

URL compression works by breaking the URL into fragments (query parameters) and replacing the fragments in the URL with shortened tokens. These shortened tokens are used by the MobileAware Interaction Server to map a request generated from the replacement URL back to the original URL.

## Examples

The following is an example of a URL of 359 characters produced by BEA WebLogic Portal:

```
/avitekfinancial/application?namespace=tracking&origin=searchResults.jsp&
event=link.clickContent&com.bea.event.type=com.bea.content.click.event&
com.bea.event.userid=null&com.bea.event.documentid=Avitek/DemoDocuments/Demo
Features
List.xls&com.bea.event.documenttype=AvitekDocs&contentId=Avitek/DemoDocume
nts/Demo Features List.xls
```

With URL compression turned on in the MobileAware Interaction Server, this URL would be reduced to 99 characters, which is a saving of 260 characters:

```
/avitekfinancial/application?2=!!3&!4=!!5&!6=!!7&!8=!!9&!10=!!11&!12=
!!13&!14=!!15&!16=!!13
```

URL compression can be configured in the `mis.properties` file. **mis.properties** file.

The table below shows sample URL Compression configuration for the MobileAware Interaction Server running against a BEA WebLogic Portal server.

**Table 7 URL Compression Properties Settings for MIS Running Against BEA WebLogic Portal**

Property	Description
<code>url.compression.store.type</code>	Defines the store type to be used. The only valid type in the MobileAware Interaction Server is <code>session</code> .  Example: <code>session</code>

<code>url.compression.token.prefix</code>	<p>The string used to prefix the compression tokens. Prefixing helps avoid clashes with uncompressed tokens which may have the same value as a compressed token.</p> <p>Default is "!!".</p> <p><b>Note:</b> The Nokia Mobile Internet Toolkit 3.1 does not support "!!".</p>
<code>url.compression.params</code>	<p>Comma separated list of query parameter names to be compressed.</p> <p>Example: namespace, event, com.bea.event.type, com.bea.event.userid, com.bea.event.documentid, com.bea.event.documenttype, contentId, origin, pageid, portletid</p>
<code>url.compression.vals</code>	<p>Comma separated list of query parameter names that have values to be compressed.</p> <p>Example: namespace, event, com.bea.event.type, com.bea.event.userid, com.bea.event.documentid, com.bea.event.documenttype, contentId, origin, pageid, portletid</p>
<code>url.compression.fail.redirect</code>	<p>The URL to which MIS will redirect a request if unrecognized compression tokens are received. This can happen, for example, if the client's session has expired and a page is refreshed or a bookmark is visited. A sensible value for this property would be the home page or login page of the site.</p> <p><b>Note:</b> the URL specified is webapp relative)</p> <p>Example from BEA WebLogic Portal deployment:</p> <pre>url.compression.fail.redirect: /avitekfinancial/application/</pre>

**Note:** When using the redirect URL for failed decompression it is recommended that content developers design JSP or XHTML pages that do not make use of, or depend on, the values of parameters passed in the URL.

## Error Handling (configuration optional)

The location of Error handler JSPs for HTTP and WAP can be configured using the parameters in the table below. The location of the JSP error handlers is a webapp relative path.

**Table 8 JSP Error Page Properties Settings**

Property	Description
<code>error.handler.jsp</code>	Location of the HTTP JSP error handler, or your own custom file. Example: <code>/errorHandler.jsp</code>
<code>error.handler.wap.jsp</code>	Location of the WAP JSP error handler, or your own custom file. Example: <code>/errorHandlerWap.jsp</code>

## Configuration Mode (configuration optional)

This configuration entry determines which mode of operation the MobileAware Interaction Server runs in.

**Table 9 Configuration Mode Properties Setting**

Property	Description
<code>operation.mode</code>	The MobileAware Interaction Server has two modes of operation: 'development' and 'production'. Setting the mode to 'development' provides detailed informative warning messages to enable content developers to tune and debug content during the development phase. By default, operation mode is set to 'production'. Example: <code>operation.mode: development</code>

## Generated URLs

The MobileAware Interaction Server automatically generates a number of URLs during content transformation. For content that the MobileAware Interaction Server produces for WAP phones, the MobileAware Interaction Server can produce 'Next' and 'Back to Top' links that contain the identifier of the required page. Similarly, when the MobileAware Interaction Server splits a form into multiple pages, the URLs generated by the MobileAware Interaction Server contain information about required form ID, current form ID and whether a form reset has been requested.

The parameter name that the MobileAware Interaction Server uses for these identifiers can be changed in the `mis.properties` file, in the event that they clash with those already used by content developers.

**Table 10 URL Generation Properties Settings**

Property	Description
<code>form.currpagenumber.paramname</code>	In delivering forms to menu-driven devices, the MobileAware Interaction Server splits large documents into numbered forms. The MobileAware Interaction Server uses the value of this property to create any URLs that reference the current page in a paginated form.  By default, this is set to <b>c_p</b> .
<code>form.nextpagenumber.paramname</code>	Defines the parameter name used by the MobileAware Interaction Server to reference the number of the next page in a paginated form.  By default, this is set to <b>form_n_p</b> .
<code>form.uniqueid.paramname</code>	Defines the parameter name for the MobileAware Interaction Server to use in generated URLs which reference the session-wide form identifier.  By default, this is set to <b>form_id</b> .
<code>url.idomid.paramname</code>	Defines the name of the parameter that the MobileAware Interaction Server uses to uniquely reference parsed documents in generating 'back to top' links.  By default, this is set to <b>page_id</b> .
<code>form.reset.paramname</code>	Defines the name of the parameter that the MobileAware Interaction Server places in URLs to indicate that a form submit is actually a 'reset'.  By default, this is set to <b>form_reset</b> .
<code>url.pagenumber.paramname</code>	In delivering content for WAP, the MobileAware Interaction Server splits large documents into numbered pages and delivers one page at a time. In so doing, the MobileAware Interaction Server must add a parameter to certain URLs so they explicitly reference an individual page of transformed content.  By default, this is set to <b>n_p</b> .

## Strict Attribute Handling and Delivery (configuration optional)

These configuration entries define whether the MobileAware Interaction Server rejects stand-alone ‘&’ symbols, and whether it delivers them in HTML content.

Example of malformed xml:

```
<a href="/url?param1=value1&param2=value">
```

Example of well-formed equivalent:

```
<a href="/url?param1=value1&amp;param2=value">
```

**Table 11 Strict Attribute Handling and Delivery Properties Settings**

Property	Description
<code>xsp.strictAttribute</code>	<p>For consistency with XHTML standards, the MobileAware Interaction Server parser is configured by default to reject stand-alone ‘&amp;’ symbols in XHTML attributes. For integration with pre-existing content and frameworks, this strictness can be switched off by setting this property to false. Allowable values are “true” and “false”.</p> <p>Example/Default: <code>xsp.strictAttribute: true</code></p>
<code>html.deliverStrictAttribute</code>	<p>When enabled, urls with query string parameters use the full XML entity reference “&amp;amp;” and thus will be delivered in the form:</p> <pre>/&lt;file&gt;?x=1&amp;amp;y=2&amp;amp;z=3</pre> <p>When disabled, these URLs take the form:</p> <pre>/&lt;file&gt;?x=1&amp;y=2&amp;z=3</pre> <p>Default: “false” (i.e. use ‘&amp;’). Allowable values are “true” and “false”.</p> <p>Example: <code>html.deliverStrictAttribute: true</code></p>

## Diagnostics Subscriptions (configuration optional)

The MobileAware Interaction Server is configured by default to send certain important diagnostics messages, such as error messages, to the application server console. It is possible to configure these and additional diagnostic messages to be sent either to the console or to a specified file.

Example diagnostics messages published to file:

```
diagnostics.startup.subscriptions.abcFile.topic: MIS.General.Startup
diagnostics.startup.subscriptions.abcFile.level: verbose|normal
diagnostics.startup.subscriptions.abcFile.filename: c:\\diagerrors.log
```

**Note:** "abc" is simply a placeholder for a unique identifier, to ensure that property names are unique. You are free to choose your own identifier.

**Note:** Each topic you subscribe to must be configured to output to a different file.

**Note:** At start-up, the specified file is overwritten, not appended to.

Example diagnostics messages published to the console:

```
diagnostics.startup.subscriptions.xyzConsole.topic: MIS.General.Startup
diagnostics.startup.subscriptions.xyzConsole.level: verbose|normal
```

By default, the MobileAware Interaction Server diagnostics are configured to publish start-up messages and diagnostic-audit messages (to track individuals connecting to diagnostics) to the console as follows:

```
diagnostics.startup.subscriptions.startupConsole.topic:MIS.General.Startup
diagnostics.startup.subscriptions.startupConsole.level:normal
diagnostics.startup.subscriptions.diagnosticsauditConsole.topic:MIS.Diagnostics.Subscription
diagnostics.startup.subscriptions.diagnosticsauditConsole.level:normal
```

Error messages could additionally be configured to publish to file as follows:

```
diagnostics.startup.subscriptions.errorsFile.topic:
diagnostics.startup.subscriptions.errorsFile.level:verbose
diagnostics.startup.subscriptions.errorsFile.filename:c:\\temp\\diagerrors.log
```

This is a special case where no topic is required.

## Table 12 Diagnostics Subscriptions Properties Settings

Property	Description
<code>diagnostics.startup.subscriptions.xxxFile.topic:</code>	Any diagnostic topic which can be selected from the MobileAware Interaction Server Diagnostics
<code>diagnostics.startup.subscriptions.xxxFile.level: verbose</code>	Specifies the level of diagnostics message required, either “verbose” or “normal”
<code>diagnostics.startup.subscriptions.xxxxFile.filename:</code>	Name of file to log this diagnostic subscription to

## Back to Top (configuration optional)

This configuration entry defines whether the “Back to Top” feature is enabled or disabled.

When enabled, a shortcut “Back To Top” link is provided on the device, which will allow the user to return directly to the top of the group based on the hierarchy of the current document. If the user then uses this enhancement to navigate to the top-level navigation card of the current document hierarchy, they are provided with a “Back To Top” link that returns them to the referrer.

**Table 13 Back to Top Properties Setting**

Property	Description
<code>backtotop.enabled</code>	Indicates whether “Back to Top” links should be used for paginated content. Default: true

## Content Length Settings

The `response.omitContentLength` configuration entry defines whether or not the MobileAware Interaction Server will set the content length in the response. By default, `response.omitContentLength` is set to false, implying that the MobileAware Interaction Server will set the content length in the response. If it is set to true the MobileAware Interaction Server will not set the content length and chunked encoding will be used to deliver content

**Table 14 Content Length Properties Setting**

Property	Description
<code>response.omitContentLength</code>	Indicates whether the content length should be included in the response from MobileAware Interaction Server  Default: false

### Disallowed Output Encodings (configuration optional)

The MobileAware Interaction Server determines from the incoming device request which character encodings will give the best rendering of content. In some circumstances, however, a device may incorrectly report its quality of support for a given character encoding, or there may be no valid transformation from the original content encoding to the preferred device encoding. Specifying a comma-separated list of encodings for the `disallowedOutputEncodings` property instructs the MobileAware Interaction Server never to deliver content in any of these encodings.

**Table 15 Disallowed Output Encodings Properties Setting**

Property	Description
<code>disallowedOutputEncodings</code>	Indicates output encodings that MobileAware Interaction Server should never use.  Example: <code>disallowedOutputEncodings: iso-8859-1, iso-8859-5</code>

### Optimizing Performance with the JSP Tag Library (configuration optional)

There are several steps involved in the MobileAware Interaction Server transformation process. Some of these steps can be bypassed to achieve optimal performance using the MobileAware Interaction Server JSP tag library.

For full details on achieving optimal performance with the JSP Tag Library, see the chapter section “Optimising Performance with the JSP Tag Library” in the MobileAware Interaction Server, BEA WebLogic Edition. The properties involved in this process are summarised below.

**Table 16 Optimizing JSP Tag Library Performance Properties Setting**

Property	Description
<code>mis.jsptaglib.passthrough</code>	Optimises processing when JSP files are known to contain only mm: and non mm- tags. Allowable values are “true” and “false”. Example: <code>mis.jsptaglib.passthrough: true</code>
<code>mis.passthrough.patterns</code>	Optimises processing when JSP files matching specified patterns are known to contain only mm: and non mm- tags. Example: <code>mis.passthrough.patterns: *.jsp</code>
<code>mis.bypass.patterns</code>	Bypasses certain processing where JSP files matching specified patterns are to be delivered only to FullBrowsers. Example: <code>mis.bypass.patterns: /pc/*.jsp</code>
<code>mis.fullbrowser.device</code>	Specifies fullbrowser device to be used with <code>mis.bypass.patterns</code> , or in unlicensed mode. Example: <code>mis.fullbrowser.device: Mozilla/5</code>

### Passthrough Pattern Configuration (configuration optional)

The following properties control which HTTP requests the MobileAware Interaction Server will act on. For performance reasons it is beneficial to be able to inform the MobileAware Interaction Server to not process any request that will not produce MMXHTML.

When an HTTP request is received, the MobileAware Interaction Server will check the URL against the "mis.patterns.url.nonmmxhtml" patterns. If it matches, the request will not be processed.

Otherwise, the MobileAware Interaction Server will check the "mis.patterns.url.mmxhtml" patterns. If these match, or the property "mis.patterns.url.unknown.mmxhtml" is set to true, the request is processed and the Content-Type is checked after the content has been produced.

The Content-Type check is done in much the same way as the url check. The Content-Type is checked against the "mis.patterns.contenttype.nonmmxhtml" property. If it matches, the content is delivered as produced. Otherwise, if either the "mis.patterns.contenttype.unknown.mmxhtml" is set to true or the "mis.patterns.contenttype.mmxhtml" pattern matches the content is transformed.

**Note:** If you need to modify or add to these lists you need to include the appropriate values from the default settings as you are overriding the property. Only remove from these lists if you are sure that is what you want to do. Configuring these properties incorrectly may cause the MobileAware Interaction Server to no longer process content.

In the following properties the "patterns" may be of the form:

XXX\* - starts with XXX, for example /images/\*

\*XXX - ends with XXX, for example \*.gif

\*XXX\* - contains XXX, for example \*/ignore/\*

**Table 17 Passthrough Pattern Properties Setting**

Property	Description
<code>mis.patterns.url.nonmmxhtml</code>	<p>Configure the list of URL patterns to NOT consider MMXHTML. If a request is received for a URL matching one of these patterns it will not be processed by MIS.</p> <p>Default:  <code>mis.patterns.url.nonmmxhtml: *.css *.gif *.jpg *.jpeg *.jpe *.wbmp *.swf *.dwt *.ico *.png *.txt *.pdf</code></p>
<code>mis.patterns.url.mmxhtml</code>	<p>Configure the list of URL patterns to consider potentially MMXHTML. If a request is received for a url matching one of these patterns it will be processed by MIS.</p> <p>Default:  <code>mis.patterns.url.mmxhtml=* .htm *.html *.jsp</code></p>
<code>mis.patterns.url.unknown.mmxhtml</code>	<p>Configure if an unknown URL should be considered potentially MMXHTML. If a request is received for a URL not matching the "mis.patterns.url.nonmmxhtml" or the "mis.patterns.url.mmxhtml" patterns, this property decides if it should be considered MMXHTML.</p> <p>Default:  <code>mis.patterns.url.unknown.mmxhtml: true</code></p>

<code>mis.patterns.contenttype.nonmmxhtml</code>	<p>Configure the list of Content-Type patterns to NOT consider MMXHTML. If a response is received with a Content-Type matching one of these patterns it will not be processed by MIS.</p> <p>Default:  <code>mis.patterns.contenttype.nonmmxhtml: application/* audio/* image/* message/* model/* multipart/* video/* text/css text/plain text/rtf text/vnd* text/xml</code></p>
<code>mis.patterns.contenttype.mmxhtml</code>	<p>Configure the list of Content-Type patterns to consider MMXHTML. If a response is received with a Content-Type matching one of these patterns it will be processed by MIS.</p> <p>Default:  <code>mis.patterns.contenttype.mmxhtml: text/html;* text/html text/tml text/tml;*</code></p>
<code>mis.patterns.contenttype.unknown.mmxhtml</code>	<p>Configure if an unknown Content-Type should be considered MMXHTML. If a request is received for a Content-Type not matching the "mis.patterns.contenttype.nonmmxhtml" or the "mis.patterns.contenttype.mmxhtml" patterns, this property decides if it should be considered MMXHTML.</p> <p>Default:  <code>mis.patterns.contenttype.unknown.mmxhtml: false</code></p>
<code>mis.patterns.contenttype.askjava</code>	<p>Configure whether content type should be ascertained by asking java to do the mapping from URL to mime-type.</p> <p>Default:  <code>mis.patterns.contenttype.askjava: true</code></p>

## BEA WebLogic Portal Settings (configuration mandatory for BEA Portal deployment)

**Note:** The following properties are to facilitate MIS Integration with BEA WebLogic Portal and Server. Do Not change these settings if running with BEA WebLogic Portal or Server.

**Table 18 BEA WebLogic Portal Properties Setting**

Property	Description
<code>compatibility.illegalState.weblogic</code>	Must be set to true if MIS deployed with BEA WebLogic Portal or Server. This is set by default when using the “Enable Multi-Channel” option in BEA WebLogic Workshop
<code>bea.portal.integrationOn</code>	Must be set to true if MIS deployed with BEA WebLogic Portal or Server. This is set by default when using the “Enable Multi-Channel” option in BEA WebLogic Workshop

## The ContentAssembly.properties File

This file contains additional configurable settings that relate to the way content appears on the screen. Most of these settings have to do with the text displayed in the automatically generated links, which aid navigation around sites being delivered to handheld devices. For example, Next, Back to:, Previous, and so on.

Unlike the other two properties files discussed in this section, this file is located in the webapp’s WEB-INF/lib/mmJSPtaglib.jar file. This file contains the default property values. If you wish to make changes to this file, unzip the jar, make a copy of the properties file, make the changes, then save it to:

```
WEB-INF/classes/com/mobileaware/i18n/resources/ContentAssembly.properties.
```

Multiple versions of this file can be created to provide locale specific property values. Which file is used depends on the language and region settings of the requesting device and the availability of a properties file matching those settings. This mechanism uses the Java Internationalization functionality that provides a standard for application designs that are adaptable to various languages and regions without engineering changes.

As an example, if you had customers in France and Germany who would be accessing your website, you would create two versions of the `ContentAssembly.properties` file and name them:

- `ContentAssembly_fr.properties`
- `ContentAssembly.de.properties`

Modify the property values to conform to the language of the country.

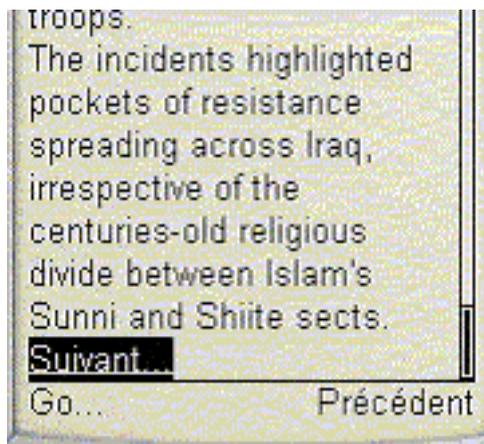
```

#
#
# Next Link
#
# The text link that is inserted by MIS
#
# during pagination of a group to take user to next
#
# page of content within the group.
#
#
#
page.group.next.link: Suivant...

#
#
# Previous Link
#
# The text link that is inserted by MIS during
#
# pagination of a group to return user to previous page #
# of the group.
#
#
#
#####
page.group.previous.link: Précédent

```

**Figure 1 Setting properties for French devices**



For more information and a list of the fully supported locales and filename extensions, see the Java documentation at [java.sun.com](http://java.sun.com).

**Table 19 Content Assembly Properties Settings**

Property	Definition
<code>page.group.next.link</code>	More... The text link that is inserted by MIS during pagination of a group to take user to next page of content within the group.
<code>page.group.previous.link</code>	Back The text link that is inserted by MIS during pagination of a group to return user to previous page of the group.
<code>form.submit.button.default</code>	Submit The default form submit button text when the user does not define it themselves.
<code>document.previous.link</code>	Back to: %previous_page_title%* The text link that is inserted by MIS during pagination that would take you to the previous page of content when in a paginated group.

<code>page.group.top.link</code>	Back to: %grp_name% The text link that is inserted by MIS during pagination of a group to take the user to the top of the page of content.
<code>table.top.link</code>	Back to:%table_name% The text link that is inserted by MIS during pagination of a group to take the user to the top of the page of content.
<code>image.alt.text.default</code>	<blank> Placed in the 'alt' text on an image when no 'alt' text is defined in an mm-img.
<code>table.link.text.default</code>	%table_name% The text link that is used to access the table data.
<code>table.empty.cell.text.default</code>	&nbsp; The text placed in an empty cell that is NOT a link.
<code>media.object.missing.alt.text</code>	<blank> Placed in the 'alt' text on an image when no media object is defined.
<code>form.next.link</code>	Next The text link included when a form is paginated.
<code>form.reset.button.default</code>	Reset The default form reset button text when the user does not define it themselves
UnNamedLink	Unnamed Link The default link text for the meta tag defining an mm-section.
Options	Options If all the Options links associated with a page do not fit on the transformed page, they will be placed on a separate page. This property specifies the text for the link to the Options page.

## The `oscache.properties` File

The `oscache.properties` file is a text file containing the configuration settings that regulate the content caching mechanism. It is located in the WEB-INF/classes folder of each your

webapps running with the MobileAware Interaction Server. It is a plain text file and can be modified in any text editor.

Normally, you do not need to modify anything in this file; however, you might want to change the directory that stores the caches. You can reset the directory by changing the `cache.path` property. By default, a cache directory called `tempCache` is created relative to the directory from which the MobileAware Interaction Server was launched.

**Note:** If you change the `cache.path` property, you must ensure that the cache has permission to write to the new directory.

## Configuring the Error Pages

The MobileAware Interaction Server provides a set of sample JSP error handling pages and images. It is recommended that you replace these with your own error handling pages and images. This error pages can be found in the top directory of the sample BEA Workshop and BEA Portal projects installed during the installation process (see the *MobileAware Interaction Server BEA WebLogic Edition Installation Guide*).

**Table 20 Sample JSP Error Handling Pages and Images**

File	Description
<code>errorhandler.jsp</code>	Defines the jsp that handles HTTP error messages
<code>errorhandlerWap.jsp</code>	Defines the jsp that handles WAP error messages

Copy the error pages to your web application directory or place them in their own directory below the `<webapp>` root if you prefer.

You will need to modify the following lines in the `mis.properties` file so that they point to the appropriate error handlers.

**Table 21 JSP Error Page Settings**

Setting	Description
---------	-------------

<code>error.handler.jsp</code>	Location of the HTTP JSP error handler, or your own custom file
<code>error.handler.wap.jsp</code>	Location of the WAP JSP error handler, or your own custom file

## Resource Administration

### Getting Started

Device profiles are stored in the MobileAware Device Database. They are managed through the Administration Console, a Java-based GUI that provides a convenient way of setting up, retrieving and modifying the attributes associated with each profile.

This chapter introduces the Administration Console.

### Quick Start

The following table introduces the basic steps in using the Administration Console.

**Table 22 Administration Console Quick Start Guide**

To...	Select
Launch the console	The MobileAware Tools Launcher Icon in BEA WebLogic Workshop or launch directly from <code>&lt;bea_home&gt;/weblogic81/mobileaware/applications/MISAdmin/MISAdministrationConsole.exe</code> or <code>MISAdministration.bin</code> .
Login	Apps > Login
Logout of the console	Apps > Logout
Close all windows	Apps > Close All
Refresh the device database	Apps > Refresh Database
Exit the console	Apps > Exit

## Logging In

1. The “Administration Console Login” window opens when you launch the application.
2. Enter the correct MobileAware Interaction Server IP address and web application address in the “Server” field, for example localhost:8080/<application>news/. (The “Server” field recalls the last four servers that the Administrator successfully connected to.)
3. If required, tick the “Password Protected” checkbox to enable the Username and Password fields.
4. If required, enter your username and password in the respective fields. As you type your password the characters appear as asterisks.
5. Click Login to display the “Administration Console” window.

## Using the System Monitor

The System Monitor displays the Free Memory available and refreshes the console.

1. Choose Apps > System Monitor

### Refreshing the Console Automatically

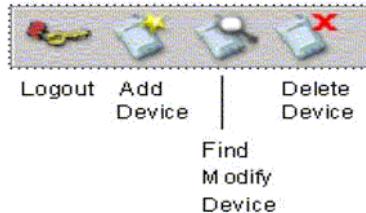
1. Set the Refresh interval (in seconds) for the Administration Console using the refresh period indicator

### Refreshing the Console Manually

1. Click Refresh to refresh the Administration Console display.

## Using the Administration Console Toolbar

The Administration toolbar provides a convenient method for accessing the administrative functions within the MobileAware Interaction Server. The following illustrates the tasks associated with each image on the toolbar.

**Figure 2 Administration Console Toolbar**

## Managing Devices

Devices have a range of input and presentation capabilities, network connectivity and levels of scripting language support.

The MobileAware Interaction Server accommodates these differences by maintaining the device database, which contains profiles describing the properties and capabilities for a range of devices on the market.

These profiles enable the MobileAware Interaction Server to tailor the presentation and delivery of content to each device. This ensures clients receive content that they can display, store and which does not take too long to convey over the network.

Device profiles are managed from within the Administration Console. From here, you can add, remove and modify devices and device attributes.

This section explains:

- The device database and device profiles
- How device profiles are used to tailor content.
- How to use the Administration Console to manage the device repository to add, delete and modify device profiles.

### About Device Profiles

Each device in the device database has an associated set of properties (attribute-value pairs) that enable the MobileAware Interaction Server to identify the requesting device in order to deliver and present the content appropriately. In the event that the MobileAware Interaction Server does not find an exact match within its profiles, it uses the attributes to determine the closest match.

Composite Capabilities/Preferences Profile (CC/PP) is a standard developed by the W3C that is used to describe device capabilities and user preferences (i.e. the delivery context). This

information can be used to develop device independent web content or applications. Based on this standard, the Open Mobile Alliance, the group that establishes open global standards for the mobile community has defined their own standard known as User Agent Profile (UAProf).

MobileAware has adopted this new standard for our device database. Currently, the database is CC/PP compliant, containing both the UAProf attribute set and a more comprehensive set of MobileAware Interaction Server proprietary device properties.

Each device is described by a set of attributes that make up a unique profile for that device. The two types of attributes are described below.

## CC/PP Attributes

Following the standard, the CC/PP compliant attributes fall into one of seven categories. Each attribute begins with a prefix that indicates into which category it falls. The following table lists these categories and gives examples of the types of attributes that they encompass.

**Table 23 CC/PP Attribute Category Prefixes and Example Attributes**

Category Prefix	Example Attributes
UAProf.BrowserUA	BrowserName FramesCapable HtmlVersion TablesCapable
UAProf.HardwarePlatform	ScreenSize ColorCapable ImageCapable Vendor
UAProf.MmsCharacteristics	MmsCcppAccept MmsMaxImage
UAProf.NetworkCharacteristics	SupportedBluetoothVersion SecuritySupport
UAProf.PushCharacteristics	Push-Accept-Charset Push-Accept-Language

UAProf.SoftwarePlatform	OSName OSVendor VideoInputEncode
UAProf.WapCharacteristics	WmlScriptLibraries WapVersion WmlDeckSize

## MobileAware Attributes

Along with the CC/PP attributes are the MobileAware proprietary attributes that complete the database. These attributes describe device characteristics that are not yet included in the standard, but describe a number of extra characteristics that can be used when tailoring content to particular devices.

## About the Device Database

### The Organization of the Device Profiles

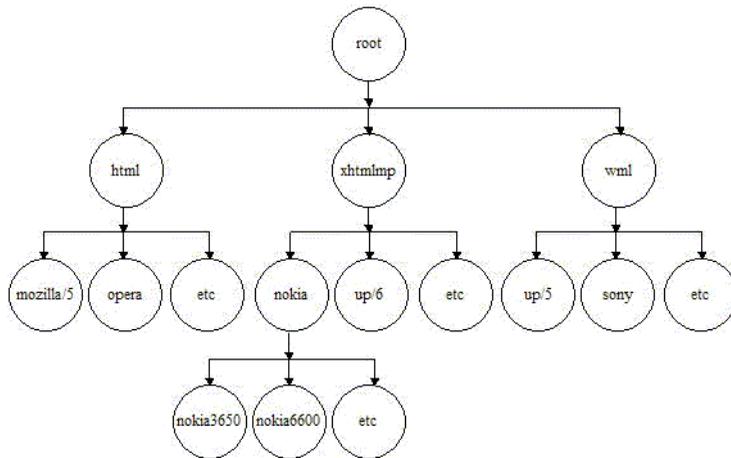
The database represents devices as a hierarchical arrangement, thus enabling devices to inherit attributes from a parent device.

The device database has three parent device classes: WML, XHTMLMP and HTML devices. When adding a new device, you can place it within one of these hierarchies or create your own parent device class.

The list of devices supported by the MobileAware Interaction Server is stored as a hierarchy or tree within the Devices section of the Administration Console.

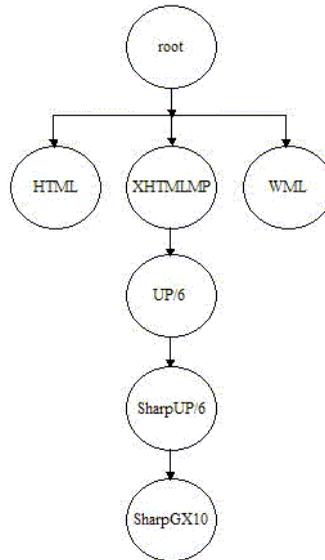
The tree is branched on the mark-up language used by the device with HTML, WML and XHTMLMP providing the main branches off the default root. Below the main branches, devices are categorised according to browser or model type, as illustrated in the device hierarchy below. An example fragment is shown below.

**Figure 3 Device Repository Hierarchy**



The hierarchy stores device attributes for the named devices detailing mark-up languages, screen sizes etc. The device name root represents the complete device tree within the MobileAware Interaction Server. All devices are subordinate to root, see example below. Tree nodes are device names, while all leaf nodes are device attributes.

**Figure 4 Device Attribute Inheritance**



A node in the device tree inherits any device attributes (markup language, screen size, etc.) from its parent. If the child node specifies values for any of these attributes, the child's values override the parent's.

## Device Recognition

### Comparing Incoming Requests to Device Attributes

When the MobileAware Interaction Server receives an end-user device request, it identifies the device using a combination of incoming request header information (which indicates the markup language of the device and often provides device model information) and stored device attributes.

The MobileAware Interaction Server achieves this by examining the details of the request and matching this request against device attributes contained in the device database.

In matching a user request against a device each level of the tree is traversed. The pattern matcher starts at the root node and attempts to traverse the tree to as deep a level (hence as specific a device match) as possible.

Each node in the tree specifies a single header and associated values that are used to differentiate it from its parent node.

The HTTPMetadataKey attribute tells the matcher to match on either Accept header or User-Agent string.

The pattern matcher will determine from the Accept header whether it is to traverse the WML branch, the XHTMLMP branch, or the HTML branch of the device hierarchy. This is defined in the HTTPMetadataString.

**Note:** The ordering of child nodes is important, as the pattern matcher will take the first match found and ignore all others.

## Multiple Header Strings

If more than one string must be present in the header, pattern matching is achieved either by using more than one level of the hierarchy with one of the strings specified in each or by combining the strings in a single node with an ampersand ('&') character. For example, a menu-driven device could require that two strings be matched: "text/vnd.wap.wml&image/vnd.wap.wbmp". Similarly, an OR comparison may be performed using the bar ('|') character.

## No Match

If an absolute match is not found a more general match is found at a higher level so that the MobileAware Interaction Server can deliver content in some format understandable by the given device.

If an unknown device sends a request, the MobileAware Interaction Server will find the closest match possible in the existing hierarchy (for example, an unknown UP 6.x browser-based phone will still match as far as UP 6.x). As such, the MobileAware Interaction Server does not need to have an exhaustive list of all devices on the market at present.

The MobileAware Interaction Server then uses the retrieved device attributes to assemble and deliver content in a manner suitable to the user's device. The content is tailored to suit the device, for example, content is restructured for smaller devices, such as WAP phones.

## Transforming Content

Once the MobileAware Interaction Server has identified a device and matched it against one in its database, it seamlessly transforms the presentation of the content to the requesting device.

This transformation is managed through the use of Device Transformation Maps (DTMs). The DTM specifies how content marked up with the MobileAware Interaction Server mobility tags is transformed to tailor the delivered page to the capabilities of the requesting device.

DTMs make it possible to accommodate new devices or upgraded versions of existing models as soon as they come on the market; transformation rules can be built quickly to take into account the new capabilities.

Each device DTM forms part of the device's profile in the database. The DTM attribute specifies the location of the transformation map to be applied to the original marked up content before it is delivered to the requesting device. The transformation map specifies how the mark-up is transformed by associating each mmXHTML/HTML tag with a Java class file that is responsible for the transformation of that tag, or by directly specifying more rudimentary transformations, such as remove or replace element.

## Tailoring Content

Device profiles enable the presentation and delivery of content to be tailored to accommodate the capabilities of the requesting device.

Within the MobileAware Interaction Server, tailoring of content takes place on three levels:

- When the MobileAware Interaction Server identifies the requesting device, it can automatically reconfigure the presentation of content to accommodate the device's capabilities, such as splitting up a large page across a number of decks on a WAP browser.
- The content author, using the conditional mobility tags, `<mm-include>` and `<mm-exclude>`, specifies how content should be altered when being delivered to different devices. For example, the length of a product description could be tailored to accommodate different-sized screens.
- The content author creates specific layouts to target different devices or device classes. Depending on the complexity of the content, the author may choose a static layout, where the dimensions (such as the number of columns and rows in a table) are fixed. Alternatively, they may choose dynamic layouts, using the delivery context API to identify the device and using JSP methods to generate the appropriate layout "on-the-fly". For example, the author can use the API to determine the width and height of a screen, and resize the table accordingly.

## Creating and Modifying Device Profiles

Device profiles are configured from within the Administration Console. The existing profiles and attributes can be modified, or new ones can be created. This can be useful for capturing more device-specific information to finely tune your content delivery for a specific purpose.

## Creating Device Profiles

Three steps are required when adding a new device profile to the MobileAware Interaction Server:

1. Complete the “Basic Details” for the device
2. Configure the standard attribute values for the device
3. Create new attributes if required

## Adding a Device

To add a device:

1. Select Device > Add Device.
2. Select the parent device class to which this device will belong.
3. Complete the details on the “Basic Device Details” tab:

**Table 24 Basic Device Details**

Field	Description
Device Name*	Type in a unique name to identify this device or device class
Display Name*	Type in the label you want displayed for this device
Description	Optionally, type in a description of this device

**Note:** \* Indicates a Required Field

4. Click Next to proceed to the next tab.

When adding a device to the database, there is a standard set of attributes that need to be configured for the new device.

## Adding a Device Attribute

To add a new Device Attribute:

1. Click Add on the Attributes tab.
2. Choose the Device attribute option.

3. In the `iName` field, type enter a name for the new attribute.
4. From the Type List, select a data type for the new attribute. If you've chosen the String data type, and want to restrict its values to a predefined list, enter a comma-separated list of values in the `iPermitted Values` field.

**Note:** The Modifiable By option should be ignored. This is a legacy option and has been deprecated.

## Configuring an Attribute

1. Select the device you wish to configure.
2. Click the Next button until you reach the Attribute Values tab.
3. Select the attribute you want to configure and double-click in the corresponding Value field.

## Modifying a Device Profile

You can add and delete attributes or change attribute values. Inherited attributes cannot be deleted: the Delete button will be grayed if you select an inherited attribute.

1. Select Device > Find and Modify Device. When the Device panel appears, select the device you want to modify. Click Next to move between tabs.
2. Click the Finish button when you are satisfied with your changes.

## Viewing an Attribute

Select the Attribute from the Attributes list and then click View.

## Deleting Devices

You can only delete devices that you have added to the device hierarchy; you cannot delete pre-installed devices.

1. Select Device > Delete Device
2. Choose the device you want to delete and click Delete.



# Appendix A – Device Attributes

This appendix lists the current attributes in the MobileAware Interaction Server device database. The listing is broken down into three major sections:.

- The CC/PP compliant attributes.
- The MobileAware proprietary attributes.
- A list of deprecated MobileAware attributes which are still supported, although their function has been replaced by a CC/PP attribute. This list will indicate which attribute should be used instead.

## The CC/PP Attributes

This section covers the major categories of CC/PP compliant attributes that allow developers to create device independent content and applications. They are listed in the database with one of the following prefixes:

- UAProf.BrowserUA  
For more information:  
<http://wapforum.org/profiles/UAPROF/ccppschem-20020710#BrowserUA>
- UAProf.HardwarePlatform  
For more information:  
<http://wapforum.org/profiles/UAPROF/ccppschem-20020710#HardwarePlatform>

- UAProf.MmsCharacteristics  
For more information:  
<http://wapforum.org/profiles/UAPROF/ccppschem-20020710#MmsCharacteristic>
- UAProf.NetworkCharacteristics  
For more information:  
<http://wapforum.org/profiles/UAPROF/ccppschem-20020710#NetworkCharacteristics>
- UAProf.PushCharacteristics  
For more information:  
<http://wapforum.org/profiles/UAPROF/ccppschem-20020710#PushCharacteristics>
- UAProf.SoftwarePlatform  
For more information:  
<http://wapforum.org/profiles/UAPROF/ccppschem-20020710#SoftwarePlatform>
- UAProf.WapCharacteristics  
For more information:  
<http://wapforum.org/profiles/UAPROF/ccppschem-20020710#WapCharacteristics>

**Table 1 CC/PP Device Attributes – UAProf.BrowserUA Prefix**

Attribute	Data Type	Example	Description
BrowserName	Literal	"Mozilla", "MSIE", "WAP42"	Name of the browser user agent associated with the current request
BrowserVersion	Literal	1.0	Version of the browser
DownloadableBrowser Apps	Literal (bag)	"application/x-java-vm/java-applet"	List of executable content types which the browser supports and which it is to accept from the network. The property value is a list of MIME types, where each item in the list is a content type descriptor as specified by RFC 2045.
FramesCapable	Boolean	Yes   No	Indicates whether the browser is capable of displaying frames
HtmlVersion	Literal	"2.0", "3.2", "4.0"	Version of HyperText Markup Language (HTML) supported by the browser

JavaAppletEnabled	Boolean	Yes   No	Indicates whether the browser supports Java applets
JavaScriptEnabled	Boolean	Yes   No	Indicates whether the browser supports JavaScript
JavaScriptVersion	Literal	"1.4"	Version of the JavaScript language supported by the browser
PreferenceForFrames	Boolean	Yes   No	Indicates the user's preference for receiving HTML content that contains frames
XhtmlModules	Literal (bag)	"XHTML1-struct", "XHTML1-blkstruct", "XHTML1-frames"	List of XHTML modules supported by the browser. Property value is a list of module names, where each item in the list is the name of an XHTML module as defined by the W3C document "Modularization of XHTML", Section 4. List items are separated by white space. Note that the referenced document is a work in progress. Any subsequent changes to the module naming conventions should be reflected in the values of this property.

**Table 2 CC/PP Device Attributes – UAProf.HardwarePlatform Prefix**

Attribute	Data Type	Example	Description
BluetoothProfile	Literal (bag)	"dialup", "lanAccess"	Supported Bluetooth profiles as defined in the Bluetooth specification [BLT].
BitsPerPixel	Number (integer)	"2", "8"	The number of bits of color or grayscale information per pixel, related to the number of colors or shades of gray the device can display.

## Appendix A – Device Attributes

ColorCapable	Boolean	Yes   No	Indicates whether the device's display supports color. "Yes" means color is supported. "No" means the display supports only grayscale or black and white. Type: Boolean Resolution: Override Examples: "Yes", "No"
CPU	Literal (string)	"Pentium III", "PowerPC 750"	Name and model number of the device CPU
ImageCapable	Boolean	Yes   No	Indicates whether the device supports the display of images. If the value is "Yes", the property CcppAccept may list the types of images supported
InputCharSet	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift_JIS"	List of character sets supported by the device for text entry. Property's value is a list of character sets, where each item in the list is a character set name, as registered with IANA.
Keyboard	Literal (string)	"Disambiguating", "Qwerty", "PhoneKeypad"	Type of keyboard supported by the device, as an indicator of ease of text entry
Model	Literal (string)	"Mustang GT", "Q30"	Model number assigned to the terminal device by the vendor or manufacturer
NumberOfSoftKeys	Number (integer)	"3", "2"	Number of soft keys available on the device
OutputCharSet	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift_JIS"	List of character sets supported by the device for output to the display. Property value is a list of character sets, where each item in the list is a character set name, as registered with IANA
PixelAspectRatio	Dimension (pair of numbers)	"1x2"	Ratio of pixel width to pixel height

PointingResolution	Literal (string)	"Character", "Line", "Pixel"	Type of resolution of the pointing accessory supported by the device
ScreenSize	Dimension (pair of numbers)	"160x160", "640x480"	The size of the device's screen in units of pixels, composed of the screen width and the screen height
ScreenSizeChar	Dimension	"12x4", "16x8"	Size of the device's screen in units of characters, composed of the screen width and screen height. The device's standard font should be used to determine this property's value. (Number of characters per row)x(Number of rows). In calculating this attribute use the largest character in the device's default font.
StandardFontProportional	Boolean	Yes   No	Indicates whether the device's standard font is proportional
SoundOutputCapable	Boolean	Yes   No	Indicates whether the device supports sound output through an external speaker, headphone jack, or other sound output mechanism
TextInputCapable	Boolean	Yes   No	Indicates whether the device supports alpha-numeric text entry. 'Yes' means the device supports entry of both letters and digits. 'No' means the device supports only entry of digits.
Vendor	Literal	"Ford", "Lexus"	Name of the vendor manufacturing the terminal device
VoiceInputCapable	Boolean	Yes   No	Indicates whether the device supports any form of voice input, including speech recognition. This includes voice-enabled browsers

**Table 3 CC/PP Device Attributes – UAProf.MmsCharacteristics Prefix**

Attribute	Data Type	Example	Description
MmsCcppAccept	Bag	"text/html"	List of content types the device supports, which can be carried inside an MMS message
MmsCcppAcceptCharSet	Bag	"US-ASCII",	The accepted character set.
MmsMaxImageResolution	String	ì120x160î	The maximum image resolution supported by the device for MMS messages.
MmsMaxMessageSize	Integer	ì1397î	The maximum size of an MMS message supported by the device
MmsVersion	Bag	ì1.0î	The version of MMS supported by the device

**Table 4 CC/PP Device Attributes – UAProf.NetworkCharacteristics Prefix**

Attribute	Data Type	Example	Description
SupportedBluetoothVersion	Literal	"1.0"	Supported Bluetooth version
CurrentBearerService	Literal	"OneWaySMS", "GUTS", "TwoWayPacket"	The bearer on which he current session was opened.
SecuritySupport	Literal (bag)	"WTLS-1", "WTLS-2", "WTLS-3", "signText", "PPTP"	List of types of security or encryption mechanisms supported by the device.
SupportedBearers	Literal (bag)	"GPRS", "GUTS", "SMS", CSD", "USSD"	List of bearers supported by the device

**Table 5 CC/PP Device Attributes – UAProf.PushCharacteristics Prefix**

Attribute	Data Type	Example	Description
Push-Accept	Literal (bag)	"text/html", "text/plain", "image/gif"	List of content types the device supports, which can be carried inside the message/http entity body when OTA-HTTP is used. Property value is a list of MIME types, where each item in the list is a content type descriptor as specified by RFC 2045
Push-Accept-Charset	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift_JIS"	List of character sets the device supports. Property value is a list of character sets, where each item in the list is a character set name registered with IANA.
Push-Accept-Encoding	Literal (bag)	"base64", "quoted-printable" "	List of transfer encodings the device supports. Property value is a list of transfer encodings, where each item in the list is a transfer encoding name as specified by RFC 2045 and registered with IANA.
Push-Accept-Language	Literal (sequence)	"zh-CN", "en", "fr"	List of preferred document languages. If a resource is available in more than one natural language, the server can use this property to determine which version of the resource to send to the device. The first item in the list should be considered the user's first choice, the second the second choice, and so on. Property value is a list of natural languages, where each item in the list is the name of a language as defined by RFC 3066

Push-Accept-AppID	Literal (bag)	"x-wap-application:wml.ua", "*"	List of applications the device supports, where each item in the list is an application-id on absoluteURI format as specified in [PushMsg]. A wildcard ("*") may be used to indicate support for any application.
Push-MsgSize	Number	"1024", "1400"	Maximum size of a push message that the device can handle. Value is number of bytes.
Push-MaxPushReq	Number	"1", "5"	Maximum number of outstanding push requests that the device can handle.

**Table 6 CC/PP Device Attributes – UAProf.SoftwarePlatform Prefix**

Attribute	Data Type	Example	Description
AcceptDownloadableSoftware	Boolean	Yes   No	Indicates the user's preference on whether to accept downloadable software.
AudioInputEncoder	Literal (bag)	"G.711"	List of audio input encoders supported by the device
CcppAccept	Literal (bag)	"text/html", "text/plain", "text/html", "image/gif"	List of content types the device supports. Property value is a list of MIME types, where each item in the list is a content type descriptor as specified by RFC 2045.
CcppAccept-Charset	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift_JIS"	List of character sets the device supports. Property value is a list of character sets, where each item in the list is a character set name registered with IANA
CcppAccept-Encoding	Literal (bag)	"base64", "quoted-printable"	List of transfer encodings the device supports.  Property value is a list of transfer encodings, where each item in the list is a transfer encoding name as specified by RFC 2045 and registered with IANA.

CcppAccept-Language	Literal (sequence)	"zh-CN", "en", "fr"	List of preferred document languages. If a resource is available in more than one natural language, the server can use this property to determine which version of the resource to send to the device. The first item in the list should be considered the user's first choice, the second the second choice, and so on. Property value is a list of natural languages, where each item in the list is the name of a language as defined by RFC 3066[RFC3066]
DownloadableSoftwareSupport	Literal (bag)	"application/x-msdos-exe"	List of executable content types which the device supports and which it is willing to accept from the network. The property value is a list of MIME types, where each item in the list is a content type descriptor as specified by RFC 2045
JavaEnabled	Boolean	Yes   No	Indicates whether the device supports a Java virtual machine
JavaPlatform	Literal (bag)	"Pjava/1.1.3-compatible", "MIDP/1.0-compatible", "J2SE/1.0-compatible"	The list of JAVA platforms and profiles installed in the device. Each item in the list is a name token describing compatibility with the name and version of the java platform specification or the name and version of the profile specification name (if profile is included in the device)
JVMVersion	Literal (bag)	"SunJRE/1.2", "MSJVM/1.0"	List of the Java virtual machines installed on the device. Each item in the list is a name token describing the vendor and version of the VM

Appendix A – Device Attributes

MexeClassmarks	Literal (bag)	"1", "3i	List of MExE classmarks supported by the device. Value "1" means the MExE device supports WAP. Value "2" means MExE device supports Personal Java, value "3" means that MExE device supports MIDP applications and value "4" means the device supports the CLI Platform. All other values should be considered reserved for use by MExE
MexeSpec	Literal	"7.02"	Class mark specialization. Refers to the first two digits of the version of the MExE Stage 2 spec
MexeSecureDomains	Boolean	Yes   No	Indicates whether the device's supports MExE security domains. "Yes", means that security domains are supported in accordance with MExE specifications identified by the MexeSpec attribute. "No" means that security domains are not supported and the device has only untrusted domain (area)
OSName	Literal	"Mac OS", "Windows NT"	Name of the device's operating system
OSVendor	Literal	"Apple", "Microsoft"	Vendor of the device's operating system
OSVersion	Literal	"6.0", "4.5"	Version of the device's operating system.

RecipientAppAgent	Literal	"BrowserMail"	User agent associated with the current request. Value should match the name of one of the components in the profile. A component name is specified by the ID attribute on the prf:Component element containing the properties of that component
SoftwareNumber	Literal	i2i	Version of the device-specific software (firmware) to which the device's low-level software conforms
VideoInputEncoder	Literal (bag)	"MPEG-1", "MPEG-2", "H.261"	List of video input encoders supported by the device
Email-URI-Schemes	Literal (bag)	"pop", "imap", "http", "https"	List of URI schemes the device supports for accessing e-mail. Property value is a list of URI schemes, where each item in the list is a URI scheme as defined in RFC 2396
JavaPackage	Literal (bag)	"com.acme.rege xp/1.1", "com.acme.help er/3.0"	(From J2EE Client Provisioning) Details about optional packages installed on the device over and above those that are part of the Java profile, and the versions of these additional packages

## Appendix A – Device Attributes

JavaProtocol	Literal (bag)	"sms/1.0", "file/1.0"	(from J2EE Client Provisioning) Details about protocols supported by the device over and above those that are part of the standard Java profile indicated and the versions of these additional protocols
CLIPlatform	Literal (bag)	"Standard CLI 2002/Compact", "Standard CLI 2002/Kernel"	The list of standard Common Language Infrastructure platforms and profiles installed in the device. Each item in the list is a name token describing the name and edition of the CLI platform specification including the name of the profile specification

**Table 7 CC/PP Device Attributes – UAProf.WapCharacteristics Prefix**

Attribute	Data Type	Example	Description
SupportedPictogramSet	Literal (bag)	"core", "core/operation", "human"	Pictogram classes supported by the device as defined in "WAP Pictogram specification".
WapDeviceClass	Literal	"A"	Classification of the device based on capabilities as identified in the WAP 1.1 specifications. Current values are "A", "B" and "C".
WapVersion	Literal	"1.1", "1.2.1", "2.0"	Version of WAP supported.
WmlDeckSize	Number	"4096"	Maximum size of a WML deck that can be downloaded to the device. This may be an estimate of the maximum size if the true maximum size is not known. Value is number of bytes
WmlScriptLibraries	Literal (bag)	"Lang", "Float", "String", "URL", "WMLBrowser", "Dialogs", "PSTOR"	List of mandatory and optional libraries supported in the device's WMLScript VM
WmlScriptVersion	Literal (bag)	"1.1", "1.2"	List of WMLScript versions supported by the device. Property value is a list of version numbers, where each item in the list is a version string conforming to Version.
WmlVersion	Literal (bag)	"1.1", "2.0"	List of WML language versions supported by the device. Property value is a list of version numbers, where each item in the list is a version string conforming to Version.

WtaiLibraries	Literal (bag)	"WTAVoiceCall", "WTANetText", "WTAPhoneBook", "WTACallLog", "WTAMisc", "WTAGSM", "WTAIS136", "WTAPDC"	List of WTAI network common and network specific libraries supported by the device. Property value is a list of WTA library names, where each item in the list is a library name as specified by "WAP WTAI" and its addendums. Any future addendums to "WAP WTAI" should be reflected in the values of this property.
WtaVersion	Literal	"1.1"	Version of WTA user agent.
DrmClass	Literal (bag)	"ForwardLock", "CombinedDelivery", "SeparateDelivery"	DRM Conformance Class as defined in OMA-Download-DRM-v1_0
DrmConstraints	Literal (bag)	"datetime", "interval"	DRM permission constraints as defined in OMA-Download-DRMREL-v1_0 . The datetime and interval constraints depend on having a secure clock in the terminal
OmaDownload	Boolean	Yes   No	Supports OMA Download as defined in OMA-Download-OTA-v1_0

## The MobileAware Attributes

These attributes are a more robust set of device characteristics describing device characteristics that are not currently covered by the CC/PP standards. They can be used to further fine-tune web content and applications.

**Table 8 MobileAware Device Attributes**

Attribute	Data Type	Example	Description
AccessKeyDisplayed	Boolean	true   false	Set to true if the browser displays the number assigned to access key beside the relevant link
AccessKeySupported	Boolean	true   false	Set to true if the browser supports access keys
AlternateLineService	Boolean	true   false	Indicates whether a device can make a voice call while keeping a data call on-line
BluetoothSupported	Boolean	true   false	Indicates whether the device is Bluetooth enabled
Brand	String	Nokia	Name of the device manufacturer
BrowserType	String	Openwave	Name of the browser
ColorGamma	Integer	1	The color gamma of the device.
ContractContiguousWhitespaces	Boolean	true   false	Set to true for those devices that do not contract insignificant whitespace when rendering markup.
DTM	String	Path to the DTM	Indicates the relevant transformation map for a device
DeliveringHTML	Boolean	true   false	Set to true if the MobileAware Interaction Server will deliver HTML to a given device. Can be used to target content at HTML devices.
DeliveringIHTML	Boolean	true   false	Set to true if the MobileAware Interaction Server will deliver IHTML to a given device. Can be used to target content and imode devices.

Appendix A – Device Attributes

DeliveringWML	Boolean	true   false	Set to true if the MobileAware Interaction Server will deliver WML to a given device. Can be used to target content at WML devices.
DeliveringXHTMLMP	Boolean	true   false	Set to true if the MobileAware Interaction Server will deliver XHTML MP to a given device. Can be used to target content at XHTML MP devices.
DeliveryType	Integer	1 or 2 or 3 or 4 where: 1 = HTML 2 = WindowsCE 3 = WML 4 = XHTML MP	Specifies the type of content that can be sent to the device
DeviceUsability	String	DeviceUsability_ MEDIUM	Describes the usability of the devices user interface
DisplayImgTextLinkSupported	Boolean	true   false	Indicates if images, text, and links can be rendered on the same line on the browser
DisplayImgTextSupported	Boolean	true   false	Indicates if images and text can be rendered on the same line on the browser
DisplaysImgTextLinkAsSingleObject	Boolean	true   false	Indicates if the device renders a <img><a href=
DisplaysMultipleImagesOnSameLine	Boolean	true   false	Indicates if the device supports multiple images on the same line
DisplaysWMLSelectAsNumberedList	Boolean	true   false	Indicates if the device renders a WML Select List as a numbered list
DownloadFunSupported	Boolean	true   false	Indicates whether Openwave Download Fun objects can be sent to the device
EMSSupported	Boolean	true   false	Indicates EMS support

EmailClient	String	POP3, IMAP4	Indicates the supported Email Clients of the device
EnableSSCSS	Boolean	true   false	Indicates whether the MobileAware Interaction Server will apply CSS on the server-side for this device
FlashSupported	Boolean	true   false	Set to true if the device supports Flash
FormSelectRenderedAsDropDown	Boolean	true   false	Indicates if the form <select> element is rendered as a drop down list
FormSelectRenderedAsLink	Boolean	true   false	Indicates if the form <select> element is rendered as a link to another card where the user makes the selection
FormSelectRenderedAsList	Boolean	true   false	Indicates if the form <select> element is rendered as a list, with all options displayed
ForwardLockContentTypeList	String	application/vnd.oma.drm.mesage	Indicates the content types supported for DRM Forward Lock
HTTPMetaDataExceptions	String	Opera, Mozilla/5, etc.	Indicates HTTPMetaDataStrings that should NOT be considered a match during device matching. Some User Agent strings contain generic values that can could potential cause a false match to occur. Filling in this field will allow device matching to progress further down the device hierarchy.
HTTPMetaDataKey	String	User-Agent Accept UA-OS	Indicates which part of the device's header contains the device's unique signature
HTTPMetaDataString	String	Nokia6210	Device's unique header string

## Appendix A – Device Attributes

HTTPPostSupported	Boolean	true   false	Set to true if the device supports the HTTP post method
HorizontalScrollBar	Boolean	true   false	Indicates if the device supports a horizontal scroll bar
IRDASupported	Boolean	true   false	Indicates whether the device supports Infrared Data Association standards for wireless transfer of data from one device to another
ImagesPlacedOnNewline	Boolean	true   false	Indicates if the device places images on a new line
ImgAsLinkSupported	Boolean	true   false	Indicates if the browser can render an image in <a href> tags as a hyperlink
ImgGIFSupported	Boolean	true   false	Set to true if the browser supports GIF format images
ImgGifAnimatedSupported	Boolean	true   false	Set to true if the browser can render animated GIFs as animations
ImgJpgBaselineSupported	Boolean	true   false	Set to true if the browser supports baseline JPGs
ImgJpgProgressiveSupported	Boolean	true   false	Set to true if the browser supports progressive JPGs
ImgLocalsrcSupported	Boolean	true   false	Set to true if the device has a locally stored image library and can access these images using the wml localsrc attribute of the img tag
ImgPNGSupported	Boolean	true   false	Set to true if the browser supports PNG format images
ImgSVGSupported	Boolean	true   false	Set to true if the browser supports SVG format images

ImgTypePref	String	.gif .wbmp	A comma delimited list (no spaces) of preferred image types for the browser, for example .gif, .wbmp
ImgWBMPSupported	Boolean	true   false	Set to true if the browser supports WBMP format images
IsFullBrowser	Boolean	true   false	Indicates large browser
IsLandscapePDA	Boolean	true   false	Set to true if a page designed with a landscape orientation is more suitable for the device
IsMenuDriven	Boolean	true   false	Indicates that a menu-driven design is most suitable for the device
IsPDA	Boolean	true   false	Indicates a PDA browser
IsPortraitPDA	Boolean	true   false	Set to true if a page designed with a portrait orientation is more suitable for the device
J2MEDownloadLimit	Integer	64000	Max size in bytes of the J2ME JAR that can be downloaded by the device
J2MESupported	Boolean	true   false	Indicates if the device supports J2ME
MLVersion	String	WML1.3 xHTML MP	Comma delimited list (no spaces) that specifies the markup languages the device supports
MMSReceiveSupported	Boolean	true   false	Set to true if the device can receive MMS messages
MMSSendSupported	Boolean	true   false	Set to true if the device can send MMS messages
MMSSupported	Boolean	true   false	Indicates if the device is MMS capable

Appendix A – Device Attributes

MP3Supported	Boolean	true   false	Indicates whether the device can handle MP3 format
MaxImageHeightPixels	Integer	21	Maximum height in pixels
MaxImageSize	Integer	2600	Maximum size of an image in bytes that can be received
MaxImageWidthPixels	Integer	50	Maximum image width in pixels
MaxObjectsInMessage	Integer	3	Maximum objects in a message
MaxTextSize	Integer	102400	Maximum Text Size
MaxWapDeckSize	Integer	2800	Maximum deck size, in bytes, that a device can receive
MexeSupported	Boolean	true   false	Indicates whether the device supports MExE
MultipartPreferred	Boolean	true   false	Indicates if the device prefers multipart content
NetworksSupported	String	GSM1900 GSM1800 GPRS	Comma delimited list (no spaces) of network technologies supported by the device
PDFSupported	Boolean	true   false	Set to true if the device supports PDFs
PreferTablesForNavList	Boolean	true   false	Indicates whether the device is able to properly support the tables created in navigational menu styling
PreferredCharsets	String	UTF-8;Q=0.8,ISO-8859-1	Indicates the preferred character sets for the device
ProtectWrappingContentTypeList	String	application/vnd.oma.drm.message	Indicates the content types the device supports Protect Wrapping for
RingtoneDownloadSupported	Boolean	true   false	Indicates if the device can download ringtones

RingtoneFormatSupported	String	midi, i-Melody	Indicates the ringtone formats supported by the device
RingtoneMonophonicSupported	Boolean	true   false	Indicates if the device can download monophonic ringtones
RingtonePolyphonicSupported	Boolean	true   false	Indicates if the device can download polyphonic ringtones
RingtonePref	String	rng,midi, amr	An ordered list of preferred ringtone formats
SMSLongMessagesSupported	Boolean	true   false	Indicates if the device can support SMS messages longer than 160 characters.
ScreenOrientation	String	Portrait Landscape	Specifies whether the device has a portrait (most devices) or landscape (communicators) orientation.
ScreenSaverSupported	Boolean	true   false	Indicates whether or not the device can support screensavers
SmartMessagingSupported	Boolean	true   false	Indicates if the device supports Smart Messaging
SupportsAbsoluteWidth	Boolean	true   false	Indicates if the device supports absolute widths on images
SupportsCSS	Boolean	true   false	Indicates whether the device supports Cascading Style Sheets
SupportsRelativeWidth	Boolean	true   false	Indicates if the device supports relative widths on images
SyncMLSupported	Boolean	true   false	Indicates whether the device has support for SyncML
TableRowsFunctionAsLink	Boolean	true   false	Indicates if the browser renders table rows as links automatically.
TextBrowser	Boolean	true   false	Indicates if the browser can only render text; i.e. cannot render images

Appendix A – Device Attributes

TextColumns	Integer	15	Maximum number of text columns that the screen can accommodate
TextFormatBigSupported	Boolean	true   false	Set to true if plain text wrapped in <big> tags is rendered in big font
TextFormatBoldSupported	Boolean	true   false	Set to true if plain text wrapped in bold <b> tags is rendered in bold font
TextFormatEmphasisSupported	Boolean	true   false	Set to true if plain text wrapped in <emphasis> tags is rendered in an emphasized font
TextFormatItalicSupported	Boolean	true   false	Set to true if plain text wrapped in italics <i> tags is rendered in italic font
TextFormatSmallSupported	Boolean	true   false	Set to true if plain text wrapped in <small> tags is rendered in small font
TextFormatStrongSupported	Boolean	true   false	Set to true if plain text wrapped in <strong> tags is rendered in a strong font
TextFormatUnderlineSupported	Boolean	true   false	Set to true if plain text wrapped in underline <u> tags is rendered with an underline
TextRows	Integer	3	Number of rows that the device screen can accommodate using the device system font
TitleRow	Boolean	true   false	Specifies whether the device has a title row
URLRequestLength	Integer	256	Maximum length of the URL request
USSDSupported	Boolean	true   false	Indicates whether or not the device supports USSD technology
UsableHeightPixels	Integer	570	Screen height excluding items like scroll bars

UsableWidthPixels	Integer	770	Screen width excluding items like scroll bars
UseTablesForNavList	Boolean	true   false	Indicates if tables should be used for Navigation list styling
UseUAProf	Boolean	true   false	Indicates if a manufacturer UAProf file is available for the device
VideoSupported	String	mpeg	Comma delimited list (no spaces) of the video formats that the device supports
VideoTypePref	String	mpeg,mp4	Ordered list of preferred video formats
ViewableHeight	Integer	30	Screen height in pixels
ViewableWidth	Integer	80	Screen width in pixels
WAPPushSISupported	Boolean	true   false	Indicates if the device supports WAP Push Service Indication
WAPPushSLSupported	Boolean	true   false	Indicates if the device supports WAP Push service loading
WAPPushSupported	Boolean	true   false	Indicates if the device supports WAP Push
WAPVersion	String	1.2.1	Specifies the version of WAP supported by the device
WMLScriptSupported	Boolean	true   false	Indicates if the device supports WML Script
WMLVersion	String	1.3	Specifies which version of WML the device supports
WTAIIInternationalPrefix	String	+00	Indicates the international prefix that should be used when specifying telephone numbers
WTAIMakePhoneCallSupported	Boolean	true   false	Indicates whether a device has phone dialing capabilities

WTLSsupported	String	WTLS_Class1	Indicates the WTLS class supported by the device
WavEncodingsSupported	String	PCM8	Indicates the supported Wav file encodings

## Deprecated Attributes

This is a list of the deprecated items in the database. These attributes are still functional for the purpose of backward compatibility although it is recommended that you use the alternative if available.

The attribute that should be used as a replacement is listed below each deprecated attribute name. Each of these new attributes should be prefixed with ‘**UAProf.**’ to form the complete name.

**Table 9 Deprecated MobileAware Device Attributes**

Deprecated Attribute	Data Type	Example	Description
AcceptHeader	String	text vnd.wap.wml image vnd.wap.wbmp	Comma delimited list (no spaces) used to specify the media types which are acceptable for the response (i.e., what can be sent to the browsing device).  Replaced by: SoftwarePlatform.CcppAccept
AudioFormatSupported	String	mid au wav mp3	Comma delimited list (no spaces) of audio formats the device is capable of supporting  Replaced by: SoftwarePlatform.CcppAccept
CDC1xSupported	Boolean	true   false	J2ME Connected Device Configuration  Replaced by: SoftwarePlatform.JavaPlatform
CLDC1xSupported	Boolean	true   false	J2ME Limited Device Configuration  Replaced by: SoftwarePlatform.JavaPlatform
CharsetSupported	String	utf8 ascii ISO8859-1	Comma delimited list (no spaces) of character sets supported  Replaced by: SoftwarePlatform.CcppAccept-Charset
ColorDepth	Int	12	Indicates the number of bits per pixel supported  Replaced by: HardwarePlatform.BitsPerPixel

## Appendix A – Device Attributes

ColorType	String	Colour	Specifies whether the screen is black & white, colour or greyscale Replaced by: HardwarePlatform.ColorCapable
DeviceClass	String	PDA FULLBROWSER WMLBROWSER	Describes the category of device Replaced by: IsPDA, IsPortraitPDA, IsLandscapePDA IsFullBrowser IsMenuDriven
EmailClient	String	POP3 SMTP	Comma delimited list (no spaces) which indicates the email protocols that the device supports Replaced by: SoftwarePlatform.Email-URI-Schemes
FoundationProfile1xSupported	Boolean	true   false	Java (CDC) profile Replaced by SoftwarePlatform.JavaPlatform
ImageFormatSupported	String	wbmp bmp gif animgif png jpeg	Comma delimited list (no spaces) of all of the image formats supported by the device eg gif,wbmp,png. Replaced by: SoftwarePlatform.CcppAccept
ImgMapTransformEnabled	Boolean	true   false	Set to true if image maps are to be transformed into links No replacement.
ImgMapTransformShowImage	Boolean	true   false	If true, images are also delivered with an image map. No replacement.

JavaPhone1xSupported	Boolean	true   false	Used by some devices with Personal Java Replaced by: SoftwarePlatform.JavaPlatform
JavaScriptSupported	Boolean	true   false	Indicates whether JavaScript is supported Replaced by: BrowserUA.JavaScriptEnabled
MIDP1xSupported	Boolean	true   false	Indicates if J2ME (CLDC) MIDP Profile Version 1 is supported Replaced by: SoftwarePlatform.JavaPlatform
MIDP2xSupported	Boolean	true   false	Indicates if J2ME (CLDC) MIDP Profile Version 2 is supported Replaced by: SoftwarePlatform.JavaPlatform
MXImageMapShowImage	Boolean	true   false	Allows you to display links in an image map on a PDA No replacement.
MXImageTypePref	String	.gif .wbmp	A comma delimited list (no spaces) of preferred image types for the browser No replacement.
MXListBoxHeight	Int	Any Integer	Default value is 6 No replacement.
MultipartSupported	Boolean	true   false	Indicates if the device can accept multipart content Replaced by: SoftwarePlatform.CcppAccept
OSVersion	String	4.22, 5.0, etc.	Indicates the version of the Operation System on the device, where applicable Replaced by: SoftwarePlatform.OSVersion

Appendix A – Device Attributes

OStype	String	AMX, PALM, etc.	Indicates the Operating System on the device, where applicable Replaced by: SoftwarePlatform.OSName
PersonalJava1xSupported	Boolean	true   false	Personal Java Specification Replaced by: SoftwarePlatform.JavaPlatform
ScreenAspectRatioPixels	String	1X1, 1X2, etc.	Pixels on most devices are higher than wide which explains why sometimes images can look distorted on browsers. The pixel aspect ratio specifies the width to height pixel ratio on a devices display Replaced by: HardwarePlatform.PixelAspectRa tio
SoundHandling	Boolean	true   false	Indicates whether or not the device has audio capability Replaced by: SoftwarePlatform.CcppAccept
TableSupported	Boolean	true   false	Indicates whether the device has table support Replaced by: BrowserUA.TablesCapable
WTAIAddPhoneBookEntry Supported	Boolean	true   false	This is part of WTAI support and allows a selected number to be saved to the devices phone book Replaced by: WapCharacteristics.WtaiLibraries

# Appendix B – web.xml Sample File

The web.xml file is found in the /WEB-INF directory of each web application folder.

<pre>&lt;?xml version="1.0" encoding="ISO-8859-1"?&gt; &lt;!DOCTYPE web-app   PUBLIC "-//Sun Microsystems, Inc.//DTD Web   Application 2.3//EN"   "http://java.sun.com/dtd/web-app_2_3.dtd"&gt; &lt;web-app&gt;</pre>	Standard web deployment descriptor.
<pre>&lt;filter&gt;   &lt;filter-name&gt;mobilityFilter&lt;/filter-name&gt;   &lt;display-name&gt;System Test   Filter&lt;/display-name&gt;   &lt;description&gt;System Test Filter&lt;/description&gt;</pre>	User-configured filter-name, display-name and description for the instance of the filter.
<pre>&lt;filter-class&gt;com.mobileaware.mcp.MobilityFilter &lt;/filter-class&gt;</pre>	The filter-class must be configured as shown.
<pre>&lt;init-param&gt;   &lt;param-name&gt;propertiesname&lt;/param-name&gt;   &lt;param-value&gt;/test.properties&lt;/param-value&gt; &lt;/init-param&gt;</pre>	(Optional) propertiesname. Defaults to "/mis.properties". Defines where the MobileAware Interaction Server will look for properties in the classpath of the webapp.

Appendix B – web.xml Sample File

<pre> &lt;init-param&gt;   &lt;param-name&gt;namespace&lt;/param-name&gt;    &lt;param-value&gt;com.mobileaware.mcp&lt;/param-value&gt; &lt;/init-param&gt; &lt;/filter&gt; </pre>	<p>(Optional) namespace. Defaults to "com.mobileaware.mcp". This is pre-pended to all session variables and servlet context variables used by the MobileAware Interaction Server in the webapp.</p>
<pre> &lt;filter-mapping&gt;   &lt;filter-name&gt;mobilityFilter&lt;/filter-name&gt;   &lt;url-pattern&gt;/*&lt;/url-pattern&gt; &lt;/filter-mapping&gt; </pre>	<p>Name of filter as specified above.</p> <p>Path of request to be handled by the filter.</p>
<pre> &lt;servlet&gt;  &lt;servlet-name&gt;DiagnosticsServlet&lt;/servlet-name&gt;   &lt;servlet-class&gt;    com.mobileaware.diagnostics.http.server.DiagnosticsServlet   &lt;/servlet-class&gt; &lt;/servlet&gt; </pre>	<p>Configuration for diagnostic servlet.</p>
<pre> &lt;servlet&gt;  &lt;servlet-name&gt;GUIRequestHandler&lt;/servlet-name&gt;   &lt;servlet-class&gt;    com.mobileaware.MIS.GUI.GUIRequestHandler   &lt;/servlet-class&gt; &lt;/servlet&gt; </pre>	<p>Configuration for the Administration Console servlet.</p>

<pre> &lt;servlet-mapping&gt; &lt;servlet-name&gt;DiagnosticsServlet&lt;/servlet-name&gt;   &lt;url-pattern&gt;/Diagnostics/*&lt;/url-pattern&gt; &lt;/servlet-mapping&gt; </pre>	<p>Diagnostics mapping, to be used from TextUI and DiagnosticsConsole.</p>
<pre> &lt;servlet-mapping&gt; &lt;servlet-name&gt;DiagnosticsServlet&lt;/servlet-name&gt;   &lt;url-pattern&gt;/private/Diagnostics/*&lt;/url-pattern&gt; &lt;/servlet-mapping&gt; </pre>	<p>Alternative version of Diagnostics mapping, protected by Application Server security.</p>

Appendix B – web.xml Sample File

<pre> &lt;servlet-mapping&gt;  &lt;servlet-name&gt;GUIRequestHandler&lt;/servlet-name &gt;  &lt;url-pattern&gt;/GUIRequestHandler/*&lt;/url-pattern&gt; &lt;/servlet-mapping&gt; </pre>	<p>Administration Console mapping. To be used from the Administration Console.</p>
<pre> &lt;security-constraint&gt;   &lt;web-resource-collection&gt;  &lt;web-resource-name&gt;SecureURLS&lt;/web-resource -name&gt;   &lt;description&gt;Private&lt;/description&gt;   &lt;url-pattern&gt;/private/*&lt;/url-pattern&gt;   &lt;http-method&gt;POST&lt;/http-method&gt;   &lt;http-method&gt;GET&lt;/http-method&gt; &lt;/web-resource-collection&gt; &lt;auth-constraint&gt;   &lt;description&gt;&lt;/description&gt;   &lt;role-name&gt;Acme&lt;/role-name&gt; &lt;/auth-constraint&gt; &lt;user-data-constraint&gt;   &lt;description&gt;SSL not required&lt;/description&gt;  &lt;transport-guarantee&gt;NONE&lt;/transport-guarantee &gt;   &lt;/user-data-constraint&gt; &lt;/security-constraint&gt; &lt;login-config&gt;   &lt;auth-method&gt;BASIC&lt;/auth-method&gt; &lt;/login-config&gt; &lt;/web-app&gt; </pre>	<p>Example of a Security constraint used with Diagnostics</p>