New Media Framework

NMF V2.0

User Guide

Revision	Date	Updated By	Description
1	12/10/2008	Brett Sheeran	First draft
2	06/11/2009	Brett Sheeran	Updated for version 2.0
3	13/11/2009	Brett Sheeran	Updated JAR lists
4	19/11/2009	Brett Sheeran	Added info on Segment Redirector tags and license.
5	28/01/2010	Brett Sheeran	Moved device data to section 4 and updated.
6	15/02/2010	Brett Sheeran	Added getLocation.jsp

1 About This Guide

1.1 Purpose

This manual is to be used as a central reference point for the New Media Framework (NMF) Java API. This guide does not describe the API methods in detail as this information is covered in the NMF JavaDoc.

1.2 Intended Audience

This manual is intended for developers and administrators of the New Media Framework.

1.3 Prerequisites

Users of this manual should be familiar with:

HTML / CSS concepts;

Java

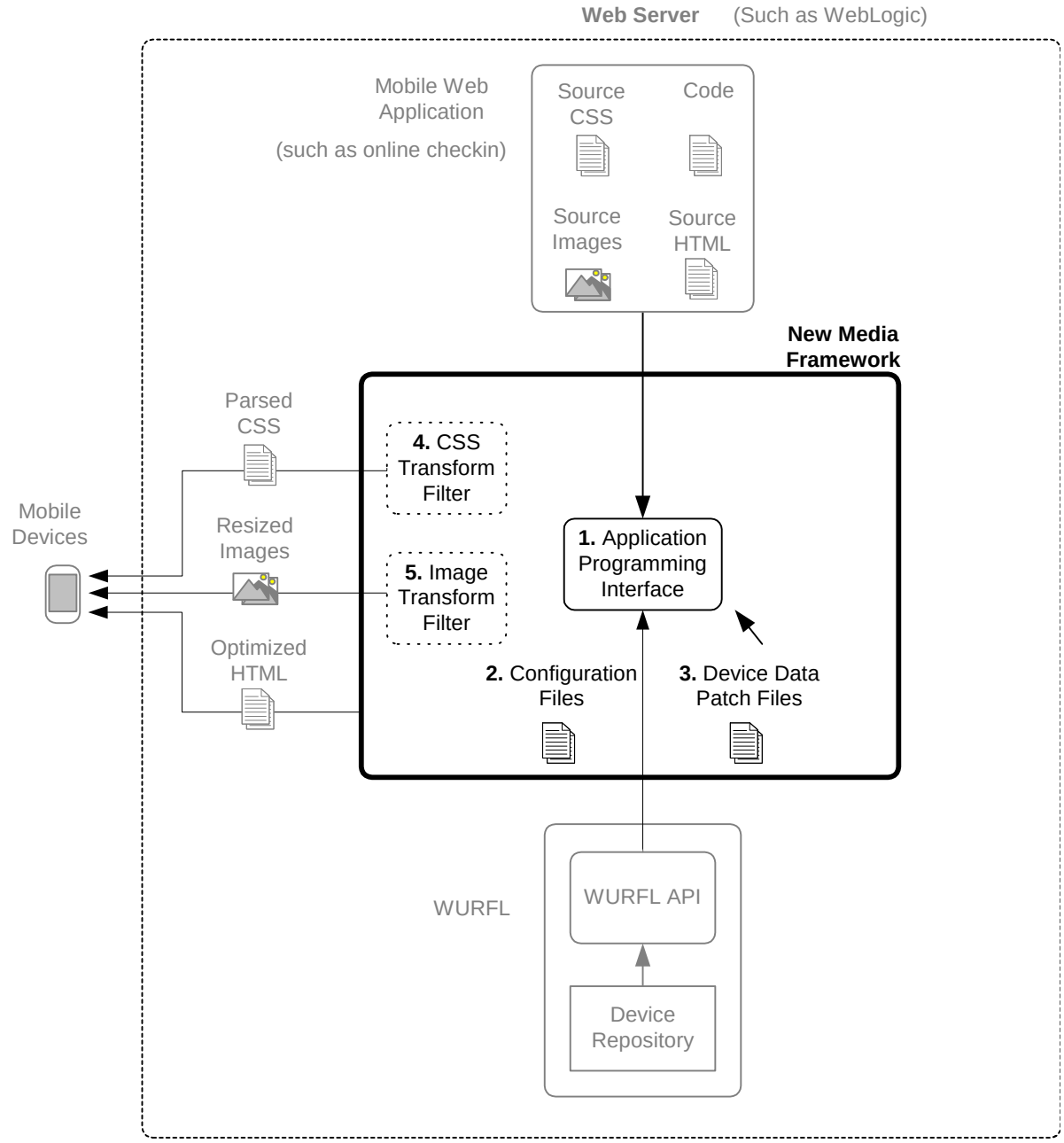
Mobile web development

2 Overview

NMF is a Java API specifically for mobile web sites. Developers will create JSPs / Servlets / Java code utilizing the NMF API. When a mobile device requests a page, NMF will render the content into mark-up suitable for that specific device. Typical variations per device include: font sizes, image sizes XHTML or HTML mark-up and JavaScript support.

NMF uses a device repository such as WURFL or MIS for obtaining device information.

The following diagram gives a high level overview of NMF.



3 Getting Started

Firstly, we suggest setting up the NMF demo on your servlet engine. Your servlet engine must be running Java 1.5 or better. Copy the NMF folder into the webapps directory. If you obtained the NMF from SVN version control, then remove the following files and folders:

.* (files and directories such as .svn, .classpath)

WEB-INF/notes.txt

WEB-INF/src

3.1 If Using WURFL

Decide the location for the following device data files:

wurfl.zip - the WURFL mobile browser file

web_browsers_patch.xml - the WURFL desktop browser file

nmf_patch.xml - optional file for any custom patches you may wish to make.

WURFLDeviceFixes.xml – the file for device specific “fixes”

The default location is WEB-INF, but you might find it more convenient to place the files elsewhere.

Download the latest wurfl.zip from http://wurfl.sourceforge.net to your data file location.

Copy web_browsers_patch.xml and nmf_patch.xml from the WEB-INF folder to your location.

Obtain the latest the latest WURFLDeviceFixes.xml from SVN location:

\content\imps\trunk\vfs\system\modules\com.nminnovation.imps\resources\

Or, request this file from New Media Innovation.

Edit web.xml:

Ensure all sections marked “Only required for MIS” ***are*** commented out and all sections marked “Only required for WURFL” are ***not*** commented out.

Change ROOT_PATH and PATCHES_PATHS to reflect the location of your wurfl.zip and web_browsers_patch.xml / nmf_patch.xml respectively.

Edit the nmf.properties (or imps.properties) file. There are a variety of parameters. At the very least you should set these values for your project:

DEVICE_REPOSITORY – in this case, to WURFL

DEVICE_FIXES_FILE – in this case the location of WURFLDeviceFixes.xml

IMG_ROOT_DIRECTORY – optional, required if downloading remote images for resizing.

If you want to remove unnecessary files, see section 11.

3.2 If Using MIS

Decide the location for the following device data files:

DeviceRepository.madr - the MIS mobile browser file

MISDeviceFixes.xml – the file for device specific “fixes”

The default location is WEB-INF, but you might find it more convenient to place the files elsewhere.

Download the latest DeviceRepository.madr file from http://mobileaware.com to your location.

Obtain the latest MISDeviceFixes.xml from SVN location:

\content\imps\trunk\vfs\system\modules\com.nminnovation.imps\resources\

Or, request this file from New Media Innovation.

Edit web.xml:

Ensure all sections marked “Only required for MIS” are ***not*** commented out and all sections marked “Only required for WURFL” ***are*** commented out.

Edit nmf.properties (or imps.properties) file. There are a variety of parameters. At the very least you should set these values:

DEVICE_REPOSITORY – in this case, to MIS

DEVICE_FIXES_FILE – in this case the location of MISDeviceFixes.xml

IMG_ROOT_DIRECTORY – optional, required if downloading remote images for resizing.

Edit mis.properties, set deviceXML.location to the location of the DeviceRepository.madr file.

Obtain a mis.license file and place in WEB-INF/classes

If you want to remove unnecessary files, see section 11.

3.3 Common Configuration for Both WURFL & MIS

Other parameters you may change include:

In the classes/log4j.properties file, set log4j.logger.com.nminnovation to the required message level and set log4j.appender.OC.File to the desired location

Also, in nmf.properties set CACHE_CONTROL_MAX_AGE and STD_CACHE_TIMEOUT as required

3.4 Copying Java Advanced Image Libraries

Different operating systems may require different versions of the JAI library files. You may download a suitable set of JAI files from Sun's web site.

You should now be able to start the NMF demo and run the demo JSPs.

4 Updating the Device Data Files

The system has 2 types of device data files, device repository files and device fix files.

Device Repository Files

The device repository file contains information on devices such as screen width. The device repository could be either MIS (Mobile Interaction Server) or WURFL (Wireless Universal Resource File). You can determine what type of repository is in use from the DEVICE_REPOSITORY parameter in the nmf.properties file located in the classes folder. The repository should be updated periodically with the latest device information files.

Device Fix Files

Device Fixes is mechanism that enables you to override specific HTML on selected devices at run time. These files will also be updated periodically, although less frequently than the device repository files. For details on the device fixes files see the Device Fixes section.

4.1 Updating WURFL Device Repository

The WURFL device repository actually consists of 3 files:

- *wurfl.zip* - a repository of data on mobile devices. We suggest updating every 5 weeks because the WURFL mobile data file is not updated on SourceForge at a specific date, but typically every 5 to 6 weeks. You may obtain the latest wurfl.zip file from http://sourceforge.net/projects/wurfl/files/WURFL/.
- *web_browsers_patch.xml* - a patch file containing data on desktop web browsers. We suggest updating every 3 months. The WURFL desktop browser data file is not updated on a specific date, but typically every few months. Multiple versions of this file can be found on the Internet. However since the content of these files varies, we recommend you obtain your copy from New Media Innovation.
- *nmf_patch.xml* – a patch file containing any custom overwrites of wurfl.zip. This file will seldom be updated

The recommended steps for updating the WURFL device repository files are as follows:

- Load the URL http://[hostname]/[context]/reloader. This step will reload the current file, but it is also a useful way to determine the location and date of the file(s). Note the dates of the files.

- Make a backup copy of the original file. We suggest giving the backup file a name like wurfl.zip.001, ...002, ...003 etc.
- Load a test page such as about.jsp to ensure the system is running correctly
- Overwrite the current file(s) with the newer version(s).
- Load the URL http://[hostname]/[context]/reloader again. Check the date has changed. The message should look like: "*Reloaded WURFL Root:C:\path\wurfl.zip:www.wurflpro.com - 2009-11-03 19:48:56*"
- Wait 10 minutes then, reload the test page

Note: if you wish to roll back to a previous version simply repeat the steps above with the older file.

4.2 Updating WURFL Device Fixes

The WURFL version of the device fixes file is called WURFLDeviceFixes.xml. This file's location can be determined from the DEVICE_FIXES_FILE parameter in nmf.properties. To update, enact the following steps:

- Find the original WURFLDeviceFixes.xml file
- Make a backup copy named WURFLDeviceFixes.xml.001, ...002, ...003 etc.
- Load a test page such as about.jsp. Preferably while emulating a device that has a device fix such as the iPhone. Make a note of the existing device fixes.
- Copy the latest file over the existing one.
- Wait 10 minutes, reload the test page. Ensure the device fixes still exist for your emulated device.

To roll-back, simply copy the older file over the new file and wait 10 minutes.

4.3 Updating MIS Device Repository

The MIS device repository consists of 1 file:

- DeviceRepository.madr. You may obtain the latest data file from MobileAware after the first Friday of every month.

The recommended steps for updating the MIS device repository are as follows:

- Determine the location of DeviceRepository.madr. You can find this by viewing the deviceXML.location value in the mis.properties file in the classes directory.
- Make a backup copy of the file. We suggest giving the backup file a name like DeviceRepository.madr.001, ...002, ...003 etc.
- Load a test page such as about.jsp to ensure the system is running correctly
- Overwrite the current DeviceRepository.madr file with the newer version.
- Wait 10 minutes then reload the test page

Note: if you wish to roll back to a previous version of DeviceRepository.madr, you must first open the “roll back” file with a zip application and save it with a later date than the currently active file. This is because MIS will only update if the file date changes to a later date.

4.4 Updating MIS Device Fixes

The MIS version of the device fixes file is called MISDeviceFixes.xml. This file’s location can be determined from the DEVICE_FIXES_FILE parameter in nmf.properties. To update, enact the following steps:

- Find the original MISDeviceFixes.xml file
- Make a backup copy named MISDeviceFixes.xml.001, ...002, ...003 etc.
- Load a test page such as about.jsp. Preferably while emulating a device that has a device fix such as the iPhone. Make a note of the existing device files.
- Copy the latest file over the existing one.
- Wait 10 minutes, reload the test page. Ensure the device fixes still exist for your emulated device.
- To roll-back, simply copy the older file over the new file and wait 10 minutes.

5 Demo JSPs

NMF comes with several JSPs that demonstrate various attributes of the NMF. The JSPs can all be reached from a page called demoindex.jsp. The code snippet below shows 2 features common to every page.

```
<%
DeviceProfile deviceProfile=DeviceProfile.getDeviceProfile(request);
if(deviceProfile.getIsXHTMLSupported()){%>
  <?xml version="1.0" encoding="UTF-8"?><!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN" "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
<%}%>
<html>
<head>
<title>My Page</title>
<%=deviceProfile.getMetaTag()%>
```

The first feature is a statement that will insert an xml tag into pages of XHTML devices.

The second feature is the deviceProfile.getMetaTag() expression. Some devices, such as iPhones and Blackberry Bolds require a metatag statement to size the page correctly. The deviceProfile.getMetaTag() method will insert a metatag where required.

The following sections describe the purpose of each JSP.

5.1 about.jsp

A JSP that will list all DeviceProfile values and device fixes for a device. It is constructed primarily with Java scriptlets. Contains logic to flush information from session and cache. Consequently, can change the DeviceProfile if doing emulation. This is a useful page to leave in your project.

5.2 beandemo.jsp

A simple page demonstrating the use of the DeviceProfile as a bean. The following commands will set the Bean and display the device usable width.

```
<jsp:useBean id="deviceProfile" class="com.nminnovation.nmf.profiles.DeviceProfile"
scope="session" >
</jsp:useBean>

UsableWidth = ${deviceProfile.usableWidth}
```

If you intend using beans you *must* enable the browser filter in web.xml. Otherwise, if the page described above is the first page viewed in a session, an error will occur.

5.3 cssdemo.jsp

A simple page that demonstrates the use of the CSSFilter. The CSS tag in cssdemo.jsp is standard HTML. However, the CSS file contains several NMF tags that NMF will replace at runtime. For further details on the NMF CSS tags, see section 10.

5.4 devicedemo.jsp

A simple page demonstrating the device profile. You can obtain a device's profile and use its values as shown below.

```
<%
DeviceProfile deviceProfile=DeviceProfile.getDeviceProfile(request);
%>
UsableWidth = <%=deviceProfile.getUsableWidth()%>
```

5.5 devicefixes.jsp

A utility page that lists the device fixes for all devices. This page is not intended for display on a mobile, it is just the list of all device fixes values such as font size metadata tags.

5.6 directionsdemo.jsp

An example of using the NMF API to generate a directions page. Figure 1 shows the result of such a search.

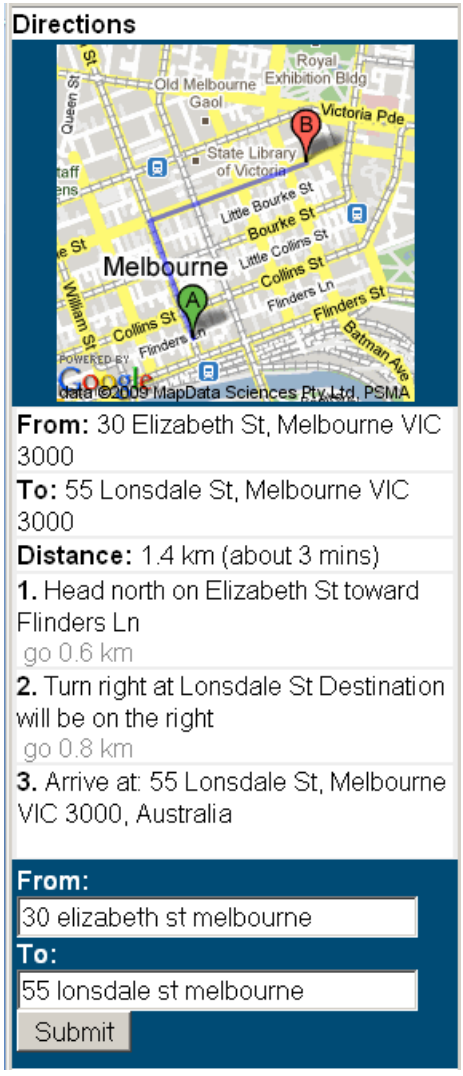


Figure 1: Directions Demo

For further detail, see the code for this JSP.

5.7 getLocation.jsp

This JSP will attempt to get location from either device, or previous search cookie or IP address lookup. If you are using the getLocation, you should install the IP address lookup database. You can download latest version from:

http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz

You should then configure the file location in imps.properties as shown below.

IP_LOCATION_FILE: /opt/maxmind/GeoLiteCity.dat

5.8 imagedemo.jsp

A simple demo that the uses an image servlet filter to resize images. This is a sample image tag.

The “{10}” sub-string is an instruction for the image servlet to render the image at 10% of usable screen width.

There is one deliberately bad image which will result in an error being logged.

5.9 imagetest.jsp

A utility page that you can use to see how an image will render for different screen widths. Figure 2 shows the result of such a test.

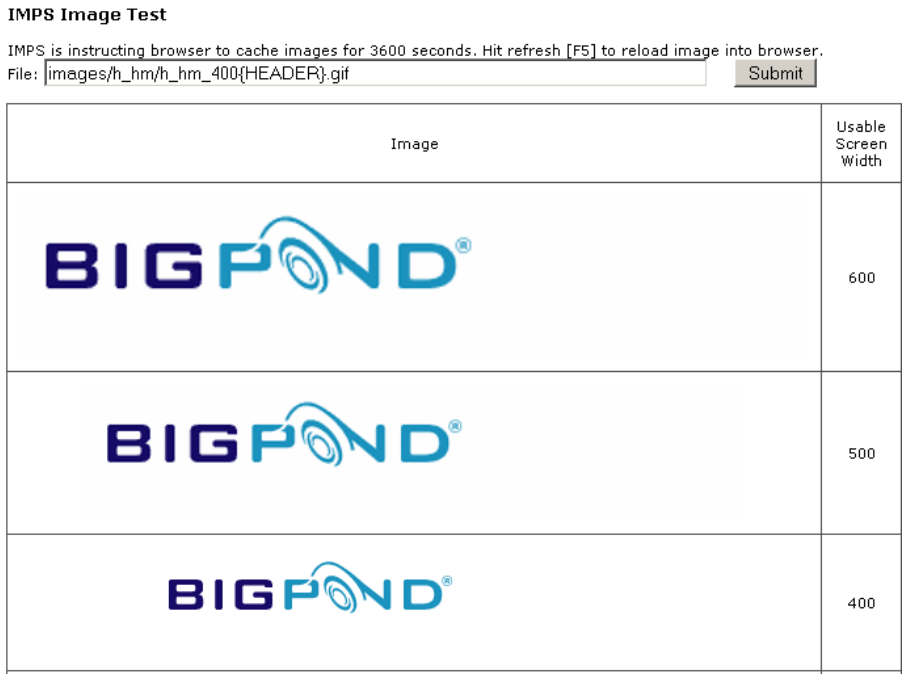


Figure 2: Image Test Page

If you are using the image manipulation features of NMF in your project, then we suggest including this JSP. If you do, change the default image file name to an existing image. The resized images are stored in the same directory as the source image. To force a resize, add the request parameter _refresh=t.

5.10 serachdemo.jsp

An example of using the NMF API to generate a business search page. Figure 3 shows an example of such a search.

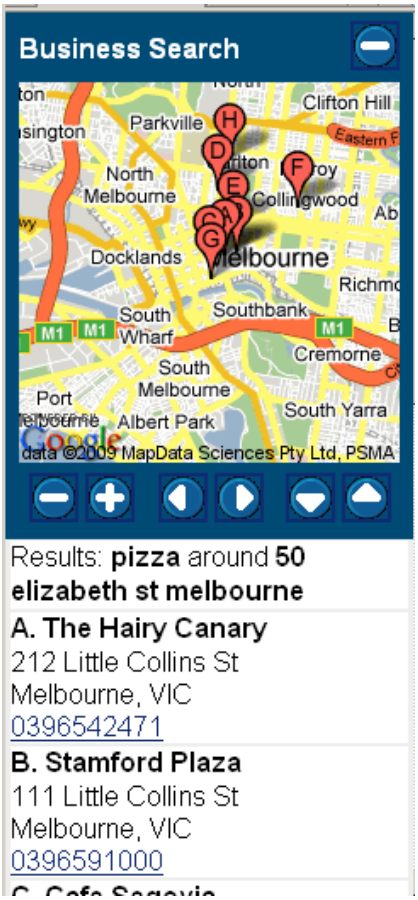


Figure 3: Business Search Page

5.11 tagsdemo.jsp

Demonstrates the various tags listed in section 9.

5.12 redirectdemo.jsp

Demonstrates redirection to one page for a mobile and another page for full browser. A developer may set either: fullbrowserurl; or mobilebrowserurl; or both.

```
<nmf:redirect
        fullbrowserurl="http://www.nminnovation.com"
        mobileurl="http://imps.nminnovation.com"
        request="<%=request%>"
        response="<%=response%>" />
```

6 Emulation URLs

By using the pd_-p parameter you may use a desktop browser to emulate a specific device. You should ensure that the first page request in the session contains this parameter. Furthermore, if you wish to then emulate another device, we recommend you close your browser then reopen to ensure a new session.

6.1 WURFL Emulation URLs

The following is a list of WURFL emulation URLs. The pd_-p value is actually the device's WURFL id, which also happens to be the UniqueID and MetaDataString in the DeviceProfile object. Some sample emulation URLs are shown below.

http://localhost:8080/base/about.jsp?pd_-p=blackberry8100_ver1

http://localhost:8080/base/about.jsp?pd_-p=blackberry9000_ver1

http://localhost:8080/base/about.jsp?pd_-p=apple_iphone_ver2

http://localhost:8080/base/about.jsp?pd_-p=nokia_6120c_ver1_sub370

6.2 MIS Emulation URLs

The following is a list of MIS emulation URLs. The pd_-p value is actually the device's unique identifier. Some sample emulation URL's are shown below.

http://localhost:8080/base/about.jsp?pd_-p=root^xhtmlmp^blackberry(xhtml)^blackberry8^blackberry8100

http://localhost:8080/base/about.jsp?pd_-p=root^xhtmlmp^blackberry(xhtml)^blackberry9^blackberry9000

http://localhost:8080/base/about.jsp?pd_-p=root^html^mozilla/5^safari^appleiphone

http://localhost:8080/base/about.jsp?pd_-p=root^xhtmlmp^nokia(xhtml)^nokia6120^nokia6120c

7 ConverterFilter Image Servlet

The ConverterFilter servlet can be used to dynamically resize images that includes the characters “{“ ”}” in the file name. For example, NMF would size a file image{90}.gif to approximately 90% of screen width.

7.1 Enabling ConverterFilter

To enable the ConverterFilter, add the following lines to web.xml

```
<filter>
        <filter-name>ConverterFilter</filter-name>
        <filter-class>com.nminnovation.imps.images.ConverterFilter</filter-class>
</filter>
<filter-mapping>
        <filter-name>ConverterFilter</filter-name>
        <url-pattern>*.jpg</url-pattern>
</filter-mapping>
<filter-mapping>
        <filter-name>ConverterFilter</filter-name>
        <url-pattern>*.gif</url-pattern>
</filter-mapping>
<filter-mapping>
        <filter-name>ConverterFilter</filter-name>
        <url-pattern>*.png</url-pattern>
</filter-mapping>
```

```
<filter-mapping>
        <filter-name>ConverterFilter</filter-name>
        <url-pattern>*.bmp</url-pattern>
</filter-mapping>
<filter-mapping>
        <filter-name>ConverterFilter</filter-name>
        <url-pattern>*.wbmp</url-pattern>
</filter-mapping>
```

7.2 Example Image Names

image.gif - will render the image in original size

image{HEADER}.gif - will render the image to a size defined in the HEADER set. Will not necessarily retain aspect ratio. For a list of image set sizes, see

Appendix A: Image Set Size.

image{90}.gif - will render the image as approximately 90% of original screen size, while maintaining aspect ratio.

7.3 Source Image Sizes

You may chose to create a source image size that is unscaled at the largest screen width breakpoint. For example, if you want an image that is 10% of screen width, you may make the source image 60 pixels wide so it unscaled at the top of the range of 600 pixels.

However, if the smaller renditions are not satisfactory, we suggest creating a source image that is approximately 3/4 to 1/2 your original source image.

7.4 img Tag Style

The tag should look like this: . The tag should not contain width or height settings.

7.5 Shortcomings of Converter Filter

The image servlet (ConverterFilter) does not currently handle the following:

Transparent GIFs, which will appear distorted

Animated GIFs, which will no longer be animated

8 Metatags

There is a meta tag device fix for the Blackberry 9000 that will insert a tag into the demo page like this:

<meta name="HandheldFriendly" content="True">

There is also a meta tag device fix for the iPhone that will insert a tag into the demo page like this:

<meta name="viewport" content="width=device-width, user-scalable=no"/>

Metatags are set in *DeviceFixes.xml.

9 Tag Libs

The following sections describe the custom tags for NMF. These tags are shown in the form <nmf:tagname…/>. But depending on your environment, they could be in the form <imps:tagname…/>.

In order to use the taglibs, you will need to insert the following markup at the top of your page.

```
<%@ taglib uri="/WEB-INF/nmf.tld" prefix="nmf"%>
```

The following sections describe use of the individual tags.

9.1 Anchor Tag

The anchor tag is used for formating links, in particular, rendering the correct format for a “click to dial” link. The anchor tag looks like this:

```
<nmf:a href="1234567"
        text="phone us"
        aclass="blank"
        style="text-decoration:none"
        request="<%=request%>" />
```

The anchor tag can be used to insert inline styles for PDA devices under MIS. More commonly, the anchor tag is used to convert a phone number to an appropriate link for the device. Examples as follows.

iPhone:

```
<a href="tel:1234567" class="blank" style="text-decoration:none">phone us</a>
```

Device that supports WTAI:

```
<a href="wtai://wp/mc;1234567" class="blank" style="text-decoration:none">phone us</a>
```

Browser without WTAI or tel support:

phone us 1234567

If the phone number does not occur in the text, then the tag will append it to the text.

9.2 Image Tag (Deprecated)

The image tag will select from a range of images based on screen width. The image tags look like this:

```
<nmf:img src="images/h_q/h_q_WIDTH.gif" imagetype="HEADER"
request="<%=request%>" extratext="style=\"border:0; margin:0\" "/>
```

Currently, the image tags only work with static sets of predefined images. The tag replaces any instances of WIDTH (or HEIGHT) with a suitable number. So, the source above could become:

```
images/h_q/h_q_220.gif
```

For a full list of images types and sizes see Appendix A: Image Set Size. However, where possible, we recommend using the image servlet filter instead.

9.3 Metatag Tag

The metattag tag will insert a metattag in the page if there is an entry in the DeviceFixes file that corresponds to the device. The metatag tag looks like this:

```
<nmf:metatag request="<%=request%>"/>
```

If the device is an iPhone for example, the resulting output would look like this.

```
<meta name="viewport" content="width=device-width, user-scalable=no"/>
```

9.4 Redirect Tag

The redirect tag will redirect either a mobile browser or a desktop browser to another page. The redirect tag looks like this:

```
<nmf:redirect
        fullbrowserurl="http://www.google.com.au"
        mobileurl="http://imps.nminnovation.com/imps/sites.html"
        request="<%=request%>"
        response="<%=response%>" />
```

If the browser is a PC browser, the server will redirect the browser to fullbrowserurl. If the browser is mobile (including PDAs) the server will redirect to mobileurl. Both fullbrowserurl and mobileurl are optional and if not included, will not redirect the respective browsers.

9.5 Segment Redirector Tag

The segment redirector tag will redirect a browser to another site “segment”. The segment redirector tag looks like this:

```
<nmf:segmentredirector
        file="../ext/segmentredirector.json"
        request="<%=request%>"
        response="<%=response%>" />
```

This tag will detect a pageName parameter and redirect the browser to its associated segment. For further details on the Segment Redirector see section 13.

9.6 Xhtmldoctype Tag

The XHTML doctype tag will insert a doctype tag if the device supports XHTML. The Xhtmldoctype tag looks like this:

```
<nmf:xhtmldoctype request="<%=request%>"/>
```

If the browser is XHTML compliant, the framework will obtain the XHTML_DOCTYPE pamater from the properties file and place it in the page. The result will probably look something like this.

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN" "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
```

10 CSSFilter

A servlet filter that will replace imps tags such as FONT_SIZE with the most suitable value for the device.

10.1.1 Enabling The CSSFilter

Add the following lines to web.xml

```
        <filter>
                <filter-name>CSSFilter</filter-name>
```

```
                <filter-class>com.nminnovation.imps.css.CSSFilter</filter-class>
        </filter>
        <filter-mapping>
                <filter-name>CSSFilter</filter-name>
                <url-pattern>*.css</url-pattern>
        </filter-mapping>
```

The following is a list of IMPS CSS tags that will be replaced at runtime

10.1.2 BG_IMAGE_X()

Used to define a horizontally repeating background image similar to the footer background image.

Sample usage:

.footer {margin:3px 0 0 0; padding:5px 5px 5px 7px; vertical-align:middle; background-color:#CEEBF7;

BG_IMAGE_X(../images/m_fbg/m_fbg{10}.gif)) IF_USE_PX_WIDTH(0)}

If the device is profile 1 or 2, IMPS will replace the placeholder with a style similar to the following:

;background-image:url(../images/m_fbg/m_fbg{10}.gif)

The image size will vary depending on the screen width. NMF does not insert the background images for profile 3 and 4 devices, but will remove the placeholder.

10.1.3 BG_IMAGE_Y()

Used to define a vertically repeating background image similar to the Link element background image.

Sample usage:

.blue {margin:3px 3px 0 3px; padding:0 0 0 3px; border: 1px solid #FFF; vertical-align:middle; background-color:#CEEBF7; font-weight:bold
BG_IMAGE_Y(../images/m_cb/m_cb{130}.jpg) IF_USE_PX_WIDTH(-1)}

If the device is profile 1 or 2, NMF will replace this placeholder with the following:

;background-image:url(../images/m_cb/m_cb{130}.jpg);background-position:left;background-repeat: repeat-y

The image size will vary depending on the screen width. NMF does not insert the background images for profile 3 and 4 devices or Blackberries, but will remove this placeholder.

10.1.4 FONT_SIZE

Used to set the default font size for a specific device.

Sample usage:

body, h2, p, td {padding:0;margin:0;border:0;font-size: **FONT_SIZE**}

Inserts the value that NMF calculates to be the best font size for the device. Either: **xx-small**, **x-small**, **medium** or **large**. Sometimes, this font will still be too small or too big. In that case, you should set a value in the device fixes.

10.1.5 IF_USE_PX_WIDTH()

Used to set a div to a specific width for certain problem devices including, Sony Ericsson k600i, k510i, k608, k750 and W800.

Sample usage:

.white {margin:3px 3px 0 3px; padding:2px 0 2px 3px; vertical-align:middle; background-color:#FFF **IF_USE_PX_WIDTH(-1)**}

This placeholder is only used for a small number of devices (such as the Sony Ericsson K600i) that do not successfully assign padding and margins. For the majority of devices NMF will remove the placeholder. If the DeviceFix attribute **SetDivWidth** is set to true (as it is for the K600i) then, NMF will replace **IF_USE_PX_WIDTH(-1)** with **width=[screen width=value]px**. So, if value=-1 and screen width=168, the replacement value would be **width=167px**.

10.1.6 USABLE_WIDTH()

Used to set a div to a specific width for *all* devices.

Sample usage:

.zz {margin:3px 3px 0 3px; padding:1px; background-color:#FFF; width:**USABLE_WIDTH(-10)**px}

NMF will replace **USABLE_WIDTH(-10)** with **width=[screen width=value]px**. So, if value=-10 and screen width=168, the replacement value would be **width=158px**.

11 Removing Extraneous Files

NMF comes with all files to use either MIS or WURFL. However, you may remove files that are not required.

11.1 Files required for WURFL

If you are using WURFL, you only require the following jar files in WEB-INF/lib:

nmf-base.jar (or nmf-core.jar)

aopalliance-1.0.jar

backport-util-concurrent-3.0.jar

commons-beanutils-1.7.0.jar

commons-collections-3.2.jar

commons-digester-1.8.jar

commons-discovery.jar

commons-lang-2.3.jar

commons-logging-1.1.jar

ehcache-1.3.0.jar

jai_codec.jar

jai_core.jar

jai_imageio.jar

json.jar

jsr107cache-1.0.jar

jstl-1.1.2.jar

log4j-1.2.8.jar

maxmind.jar

mlibwrapper_jai.jar

oscache-2.4.1.jar

rome-1.0RC1.jar

spring-aop-2.5.6.jar

spring-beans-2.5.6.jar

spring-context-2.5.6.jar

spring-context-support-2.5.6.jar

spring-core-2.5.6.jar

spring-web-2.5.6.jar
standard-1.1.2.jar
web.jar
wurfl-1.0.jar
wurfl-reloader.jar
xom-1.0.jar

If you are not using the Spring configuration for WURFL, you may remove:
aop*.jar files
spring*.jar files
WEB-INF\jmx-ctx.xml

If using WURFL, you may also remove the following MIS related files:
WEB-INF\classes\mis.* files
WEB-INF\MISDeviceFixes.xml

11.2 Files Required For MIS

If you are using MIS, you only require the following jar files in WEB-INF/lib:
nmf-base.jar (or nmf-core.jar)
archiver.jar
ccpp.jar
classes12.jar
commons-beanutils-1.7.0.jar
commons-digester-1.8.jar
commons-discovery.jar
commons-logging-1.1.jar
jai_codec.jar
jai_core.jar
jai_imageio.jar
jdom.jar
json.jar
jython.jar
log4j-1.2.8.jar
maxmind.jar

mcpfilter.jar

mlibwrapper_jai.jar

mmJSPtaglib.jar

mssql-jtds-1.2.jar

mysql-connector-java-5.0.7-bin.jar

ojdbc14.jar

oscache-2.4.1.jar

pg73jdbc2ee.jar

rome-1.0RC1.jar

sac.jar

web.jar

xom-1.0.jar

You may also remove the following WURFL related files:

WEB-INF\wurfl.* files

WEB-INF\jmx-ctx.xml

WEB-INF\ehcache.xml

WEB-INF\nmf_patch.xml

12 Device Fixes

12.1 Overview

Device Fixes is mechanism that enables you to override specific HTML on selected devices at run time. The reason we created these Device Fixes is because there are a number of undesirable behaviours observed on certain browsers that we cannot predict programmatically.

The following sections describe the purpose of each fix. The Device strings are taken from the MIS Device repository and would not be the same for other device repositories such as WURFL.

12.2 BackgroundImageXWidth

Purpose: To override NMF's automatic Footer background image selection for a specific device. Usually, NMF will successfully select a Footer background image with a taller height that the footer slot. However, a few devices have one of the following issues:

Occasionally, a footer background image is not tall enough for a particular device.

Some devices do not handle background images correctly and consequently may display unwanted behaviour such as side-scroll.

In this case, the BackgroundImageXWidth device fix may be used to override the NMF selected background image with a larger image or no image at all (not_set).

Acceptable values: not_set, 18, 25, 34, 42. These are the image widths which incidentally, are half the height.

Example Issue	Solution
The footer background image for a Nokia 6101 is too short. So, we want to increase the image size to 34 pixels.	DeviceFix: Device: nokia6101 Fix: **34**
The Nokia 6670 cannot display background images correctly. So, we want to remove the Preview background image device for “nokia6670”	DeviceFix: BackgroundImageXWidth **\nokia6670** Fix: not_set

12.3 BackgroundImageYWidth

Purpose: To override NMF’s automatic selection of background images for a specific device. Usually, NMF will successfully choose a Preview background image with a width greater than the screen width. However, a few devices have one of the following issues:

A user may switch the device from portrait to landscape mode in which case, the background image is no longer wide enough for the page.

Some devices do not handle background images correctly and consequently may display unwanted behaviour such as side-scroll.

In this case, the BackgroundImageYWidth device fix may be used to override the NMF selected background image with a wider image or no image at all (not_set).

Acceptable values: not_set, 112, 120, 154, 168, 185, 220, 280, 340, 400, 500, 600, 650

Example Issue	Solution
The JasJam V2 has both portrait and landscape mode. In landscape mode the Preview background images are not wide enough. So, we want to increase the width of the Preview background images for “jasjam&v2” to 340 pixels.	DeviceFix: Device: jasjam&v2 Fix: 340
The Nokia 6670 forces the page width to the width of the background images resulting in side-scroll. So, we want to remove the Preview background image device for “nokia6670”	DeviceFix: BackgroundImageYWidth **\nokia6670** Fix: not_set

12.4 BaseMarkup

Some devices (Blackberries in particular) ignore padding and margins in divs. When BaseMarkup is set to true, NMF will insert <hr/> tags between slots to compensate.

Acceptable values: true

Example Issues	Solutions
No Blackberry device displays any margins between slots.	DeviceFix: BaseMarkup**Blackberry*** Fix: true

12.5 DisplayTabs

Purpose: To force tabs on or off. Otherwise, NMF will display tabs if the device is profile 1 or 2 and the usable screen width is 128 pixels or more. If the device is profile 3 or 4 or the devices usable screen width is less than 128 pixels, then NMF will display a Link element for each *active* tag.

However some devices that should display tabs render the tabs so badly that we use the DisplayTabs device fix to turn the tabs off.

Other devices that have a usable screen width of slightly less 128 pixels but are quite capable of displaying the tabs, so we use the DisplayTabs to turn the tabs on.

Acceptable values: true, false

Example Issues	Solutions
The nokia6280 displays a lot of space under the tab images. So, we want to force the tabs off for this device.	DeviceFix: DisplayTabs**nokia6280** Fix: false
The Sony Ericsson K500i has a usable screen width of 124 pixels, but could still display the tabs. So, we want to force the tabs on.	DeviceFix: TabDisplay**k500i** Fix: true

12.6 FontSize

Purpose: To override NMF's automatic font selection for a specific device. Usually, NMF will determine font size based on device capabilities such as screen width and number of columns. However for a few devices, NMF's selection algorithm still generates fonts that are too small or too large. In this case, Content Operations may use FontSize to override the NMF selected font.

Acceptable values: xx-small, x-small, small, medium and large

Example Issue	Solution
The fonts on the Nokia 7370 are x-small and look tiny. So, we want to set the font for a device to a larger size.	DeviceFix: Device: nokia7370 Fix: **small**

12.7 HeaderAdd1pxImage

Purpose: insert a 1 pixel gif into the header element. This fix was introduced because of a bug in certain Nokia's that would result in these devices not displaying the first image in the page

Acceptable values: true

Example Issue	Solution
The Nokia 6250 does not display the header image on any page, yet when we emulate the 6250 the emulating browser does display the header image. So, we want to insert a 1 pixel image at the top of the page to force the Nokia 6250 to display the header image.	DeviceFix: Device: nokia6250 Fix: true

12.8 JavaScriptDisabled

Purpose: To override turn off JavaScript for devices that the Device Repository states support JavaScript, but that in practise as issues, or disable because MIS causes problems with JavaScript.

Acceptable values: true

Example Issue	Solution
The MIS device repository states that the Samsung A411 supports JavaScript, but MIS strips some of the JavaScript out.	DeviceFix: Device: SAMSUNG-SGH-A411 Fix: true

12.9 InsertLinkStyle

Purpose: To insert an inline style for an anchor tag when the device does not support inheritance of styles.

Acceptable values: true

Example Issue	Solution
There is a class in the CSS like this: .link1 a {color:#ffffff} However, the Samsung R460 browser does not render links inside as white. If InsertLinkStyle is set, then the link will be rendered like this: Weather	DeviceFix: Device: *r460_generic.xml Fix: true

12.10 MetaTag

Some devices require a meta tag in the header to ensure that the page is sized correctly.

Acceptable values: any valid header tag

Example Issue	Solution
The iPhone is displaying pages that are far too small.	DeviceFix: Device: **iPhone*** Fix: <meta name="viewport" content="width=device-width, user-scalable=no"/>

12.11 Profile

Purpose: To override NMF's automatic selection of a device profile number. The reason, a few devices have one of the following issues:

A device cannot handle the CSS assigned to it and must have its profile number reduced to 4.

Some profile 1 and 2 devices cannot handle tabs and must be reduced to profile 3.

In this case, the Profile device fix may be used to override the NMF selected background image with a wider image or no image at all (not_set).

Acceptable values: 1, 2, 3, 4

Example Issue	Solution
The LG-KG800 is (by default), profile 3, but cannot handle CSS correctly.	DeviceFix: Device: **LG-KG800** Fix: 3
The Nokia 6666 (not a real device) is by default profile 2, but cannot render tabs without spaces underneath.	DeviceFix: Device: nokia6666 Fix: 3
The Nokia 7777 (not a real device) should be profile 2, but because the MaxWapDeckSize is not correct, NMF is displaying it as profile 3.	DeviceFix: Device: nokia7777 Fix: 2

12.12 SetDivWidth

Purpose: To set the div widths for devices that do not resize nested tags correctly. The reason, some older Sony Ericsson devices do not handle nested divs well:

Acceptable values: true

Example Issue	Solution

The Sony Ericsson K600i makes nested divs 100% of screen width in spite of any padding	DeviceFix: Device: SonyEricssonK600i Fix: true

12.13 SetScreenWidth

Purpose: To set the width of the main div to a specified value. The reason, some devices render the content wider than the page.

Acceptable values: A usable screen width in pixels

Example Issue	Solution
The Samsung R810 is rendering content much wider than the usable screen width. The Device Fix will force width to be 350 pixels.	DeviceFix: Device: **samr810** Fix: 350

12.14 SingleColumnOnly

With NMF, it is possible to use floating divs to display multiple columns for wider screen devices. However, not all devices support floating divs. SingleColumnOnly can be used to force the device to display a single column.

Acceptable values: true

Example Issue	Solution
A Samsung A701 does not display floating divs side by side.	DeviceFix: Device: **SGH-A701** Fix: true

12.15 TelLinkSupported

Apple devices support phone links in the format href=tel:12345678 rather than wtai format.

Acceptable values: true

Example Issue	Solution
The iPhone does not support wtai links, but tel format instead.	DeviceFix: Device: ***iPhone*** Fix: true

12.16 UseUILibrary

Indicates that the device will support a UI Library such as iWebKit or CiUI

Acceptable values: true

Example Issue	Solution
The iPhone supports iWebKit which we want to use for page animation..	DeviceFix: Device: ***iPhone*** Fix: true

12.17 ViewPortWidth

Deprecated: recommend using meta tag instead.

Purpose: insert a viewport Meta Tag for iPhones and iPods to ensure the default screen width is of acceptable size. Currently, does not actually use the width in the viewport tag, but this may change in future.

Acceptable values: typically 320

Example Issue	Solution
An iPhone displays pages at less than full screen width	DeviceFix: Device: **iPhone** Fix: 320

13 Simple Page Example

The following is a very simple page example.

```
<%@ page import="com.nminnovation.nmf.profiles.DeviceProfile" %>
<%@ taglib uri="/WEB-INF/nmf.tld" prefix="nmf"%>
<nmf:xhtmldoctype request="<%=request%>"/>
<%DeviceProfile deviceProfile=DeviceProfile.getDeviceProfile(request);%>
<html>
<head>
<nmf:metatag request="<%=request%>"/>
</head>
<body>
Device is: <%=deviceProfile.getDeviceUniqueName()%><br/>
Screen width: <%=deviceProfile.getUsableWidth()%>
</body>
```

```
</html>
```

This example illustrates use of the xhtmldoctype and metatag tags. This example also illustrates how you might use the API to obtain device data. For father details on the API, see the NMF JavaDocs.

14 Segment Redirector

The Segment Redirector is a class that can be used to redirect different browsers to different URLs from a single URL. For example, you might have a segment of your site devoted to iPhone optimized pages. Using the segment redirector, you could have one source URL (for example redirect.jsp?pageName=news) redirect iPhones to one page and all other browsers to another page. You configure redirection in a JSON file similar to the file listing shown below.

```
{

"Segments": [
      {
            "SegmentName" : "iphone",
            "DeviceIds": ["*iphone*", "*ipod*"]
      },
      {
            "SegmentName" : "blackberry",
            "DeviceIds": ["blackberry*"]
      },
      {
            "SegmentName" : "default",
            "DeviceIds": []
      }
],

"Pages": [
      {
            "PageName" : "news",
            "SegmentUrls": [
                  {"Segment": "iphone",       "Url": "iphone/news.html"},
```

```
                {"Segment": "blackberry",       "Url": "blackberry/news.html"},
                {"Segment": "default",      "Url": "default/news.html"}
            ]

        },
        {
            "PageName" : "sport",
            "SegmentUrls": [
                {"Segment": "iphone",       "Url": "iphone/sport.html"},
                {"Segment": "blackberry",       "Url": "blackberry/sport.html"},
                {"Segment": "default",      "Url": "default/sport.html"}
            ]

        },
        {
            "PageName" : "default",
            "SegmentUrls": [
                {"Segment": "iphone",       "Url": "iphone/index.html"},
                {"Segment": "blackberry",       "Url": "blackberry/index.html"},
                {"Segment": "default",      "Url": "default/index.html"}
            ]

        }
]

}
}
```

Devices Section

In the Devices section there are three segments: iphone, blackberry and default (the segment for all devices not otherwise listed. For each segment there is a list of devices. You may list any number of devices. The DeviceIds are actually the HttpMetaData string from the device profile. You may use wildcards such as “*” and “?”. The default segment is not actually used by the code and is only listed to aid understanding.

Pages Section

The pages section lists the pages that might be requested by a parameter like pageName=news. If no pageName parameter is specified, then the segment redirector will choose the default page.

Updating the Configuration File

You may update the configuration file at runtime. However, the framework may take up to 10 minutes to detect the new file. We strongly recommend you run a JSON validator over the file before deployment.

Simple Code example

The following code shows how you might utilize the Segment Redirector.

```
SegmentRedirector segmentRedirector =
SegmentRedirector.getInstance("../ext/segmentredirector.json");

String pageName = request.getParameter("pageName");

String url = segmentRedirector.getSegmentUrl(pageName, request);

response.sendRedirect(url);
```

You may also use the segmentredirector tag as described in section 9.5.

15 General Troubleshooting

This section describes some general issues and possible solutions.

15.1 Fonts to Big or Small

The fonts on my device are too big/small.

Issue: NMF replaces any instances for the FONT_SIZE tag in CSS with an estimated size such as xx-small, x-small etc based on screen width and columns. Assuming such a tag does exist in your CSS then NMF font size calculation is not optimal for your device.

Solution: add a FontSize fix as described in section 12.6.

15.2 Background Images do Not Render Correctly

The background images are too small

Issue: Many newer devices can be viewed in both portrait and landscape mode. Often, the background images are not wide enough for landscape.

Solution: Add a device fix for either a larger background image or no background image as described in section 12.3.

15.3 All Device Profiles the Same

If all device profiles are the same, often indicated by all images being too large, check your license file to ensure it is valid.

16 License File

The license file is located at WEB-INF/classes/nmf.license. You can obtain a valid license file from New Media Innovation.

17 Device Profiles

Depending on device capability, NMF will categorise devices as either Profile 1, 2, 3 or 4. An application may utilise these profiles to will deliver its richest content/mark-up to Profile 1 and most basic content/mark-up to Profile 4. You may also configure a device to a lower quality profile if that device does not render correctly at its default profile. For example, an application may not display background images for profile 3 and 4 devices.

17.1.1 Profile 1

NMF designates all 3G devices as Profile 1.

NMF will designate all non-3G devices as either Profile 2, 3 or 4.

17.1.2 Profile 2

NMF designates non-3G devices with deck sizes of 9 kb or greater as Profile 2.

17.1.3 Profile 3

If a device has a maximum WAP deck size of greater than or equal to 5 kb and less than 9 kb.

17.1.4 Profile 4

If the device has a maximum WAP deck size of less than 5kb, or the device is WML.

18 Glossary

Term	Description

NMF	New Media Framework. A Java API for developing mobile websites.
MIS	Mobile Interaction Server. Device Repository and rendering software produces by MobileAware
WURFL	Wireless Universal Resource File. An open source device repository.

19 Appendix A: Image Set Size

19.1 Image Size Table

The following table lists the image sizes for various image types. The names of images should reflect contain their actual widths. For example, a set of CALL_TO_ACTION_ICON(s) might be called:

icon_23.gif

icon_18.gif

icon_15.gif

icon_13.gif

For most images, NMF will select the largest image that does not exceed the Usable Screen Width. The exception to this rule is the background images which are generally larger than the Usable Screen Width.

If you are creating a set of static external or internal images (with WIDTH or HEIGHT in the name) they should generally conform to theses sizes.

If you create an internal dynamic image, then NMF will render the image to the selected sizes.

Image Size table

Usable Screen Width	Image Width	Image Height
BACKGROUND		
600	650	15
500	600	15
400	500	15
340	400	15
280	340	15
220	280	15
185	220	15
168	185	15
154	168	15
120	154	15
112	120	15
BANNER_PROMOTION		
340	330	79
280	270	64
220	210	50

185	175	42
168	158	38
154	144	34
120	110	26
96	86	20
84	74	18
CATEGORY_ICON		
400	25	25
220	20	20
168	18	18
96	15	15
GRID		
320	75	90
220	50	60
80	40	45
HEADER and SCREENWIDTH_BREAKPOINTS		
600	600	121
500	500	101
400	400	81
340	340	69
310 (screenwidth only)	310	
280	280	57
220	220	45
185	185	37
168	168	34
154	154	31
120	120	24
112	112	24
96	96	24

84	84	24
PORTRAIT		
600	190	270
500	160	227
400	125	177
340	105	149
280	85	121
80	70	99
TAB		
340	87	40
220	53	28
168	48	24
140	44	20
96	38	16
FOOTER_BACKGROUND		
220	42	84
168	34	68
96	25	50
SQUARE_THUMBNAIL_PREVIEW		
600	160	160
500	140	140
400	120	120
340	100	100
280	80	80
185	60	60
125	40	40
LANDSCAPE_THUMBNAIL_PREVIEW		
600	240	160

500	210	140
400	180	120
340	150	100
280	120	80
185	90	60
125	60	40
CALL_TO_ACTION_ICON		
400	23	23
220	18	18
168	15	15
96	13	13
PREVIEW		
400	46	46
220	36	36
168	30	30
96	26	26

20 References and Suggested Reading

New Media Framework Java Doc