



# INFUZE APACHE MODULE USER GUIDE

## Support

The [ScientiaMobile Support Forum](#) is open to all WURFL users, both commercial license holders and evaluation users. It represents the combined knowledge base for the WURFL community. Commercial licensees are invited to post questions in the forum using the account to which their licenses are associated. This may mean faster handling of those posts by ScientiaMobile's personnel.

For commercial license holders, there are tiered support levels to address a variety of business support needs. After logging into your account, commercial licensees with support options can access the [Enterprise Support](#) portal to post tickets. These tickets will receive expedited attention.

To inquire about support plans, use our [License Inquiry](#) or our [General Inquiry form](#).

## Update Notifications

If you would like to be notified of our API updates, major data updates, and other technical changes, please [subscribe](#) to our ScientiaMobile Announcements list

# WURFL InFuze Module for Apache: User Guide

This document is about the Apache server and how you, a developer or a system administrator, would install and configure the WURFL Module for Apache on Unix (Linux) systems.

## Installing libwurfl

In order for the Module to work it is **ESSENTIAL** that the libwurfl library is installed on your system. libwurfl is provided in your Customer Vault/FileX.

If you have not already installed libwurfl, instructions can be found [here](#). Release notes for each API can be found [here](#).

## Install WURFL Apache Module on Ubuntu

Run the following commands to install the latest Apache software:

```
sudo apt-get update
sudo apt-get install apache2 apache2-threaded-dev
```

Download and install the WURFL Apache module deb package:

```
sudo dpkg -r apache-mod-wurfl
sudo dpkg -i mod_wurfl-1.9.0.0.Apache.2.2.32.x86_64.deb
```

## Install WURFL Apache Module on RedHat/Fedora/CentOS

Run the following command to install the latest Apache Cache software:

```
sudo yum update
sudo yum -y install openssl-devel
sudo yum -y install pcre httpd
```

Download and install the WURFL Apache module rpm package:

```
sudo rpm -e apache-mod-wurfl
sudo rpm -i mod_wurfl-1.9.0.0.Apache.2.2.32.x86_64.rpm
```

## Install WURFL Apache Module on Mac OS X

WURFL Apache module for Mac OS X is distributed as a tar.gz package containing the mod\_wurfl.so library and the sample configuration file wurfl.conf.

Unpack it and copy

- mod\_wurfl.so in Apache modules folder (typically /usr/libexec/apache2)
- wurfl.conf in Apache configuration folder (typically /private/etc/apache2/other)

# Configuration Guide

Apache is configured by placing directives in plain text configuration files. The main configuration file is usually called `httpd.conf` and includes the WURFL Apache Module configuration file `wurfl.conf`.

The installation package will place a sample `wurfl.conf` file in `/usr/share/wurfl` folder.

The `wurfl.conf` contains a `LoadModule` directive and a `<IfModule></IfModule>` section including WURFL module configuration directives.

To activate WURFL Apache module you have to customize the `<IfModule></IfModule>` section of `wurfl.conf` file and copy it in the Apache modules configuration folder (whose path depends on which Linux/Mac OSX distribution and Apache version you are installing the module).

Below is an example snippet of the `wurfl.conf` `<IfModule></IfModule>` section for WURFL setup:

```
<IfModule wurfl_module>
# -- WURFL root definition. User MUST specify this path in order to make WURFL engine correctly start.
WurflRoot /usr/share/wurfl/wurfl.zip

# -- WURFL Updater allows seamless update of WURFL engine with new data downloaded from Scientiamobile.
# -- Put your personal updater url taken from Scientiamobile customer Vault.
# -- WURFL file should be either .zip or .xml.gz and match WurflRoot file type
# -- Valid values for the updater check frequency (how often the updater checks for any new WURFL data file
# -- to be downloaded and used by the engine) are DAILY, WEEKLY
# -- Updater log file (wurfl-updater.log) may be found in "WurflRoot" folder. The folder and wurfl.zip file must be
# -- writable by Apache process
#WurflUpdater https://data.scientiamobile.com/xxxxx/wurfl.zip DAILY

# -- WURFL patches definition (as much as needed, patches will be applied in the same order as specified in this conf file)
#WurflPatch /path/to/first/patch.xml

# -- WURFL UA priority: one of the following (default is WurflUserAgentPriorityOverrideSideloadedBrowserUserAgent)
#WurflUserAgentPriorityOverrideSideloadedBrowserUserAgent
#WurflUserAgentPriorityUsePlainUserAgent

# -- WURFL cache: one of the following
#WurflCacheNull
#WurflCacheLRU 10000

# -- WURFL user requested capabilities (as an example, this is not a complete list)
#WurflRequestCapability is_console
#WurflRequestCapability is_tablet
#WurflRequestCapability is_wireless_device

# -- WURFL user requested virtual capabilities (as an example, this is not a complete list)
# -- Since WURFL API version 1.7.1.0, virtual capabilities are no longer injected by default
# -- and have to be explicitly specified.
#WurflRequestCapability advertised_device_os
#WurflRequestCapability is_android

# -- WURFL user requested properties
# -- Since WURFL API version 1.8.0.0, WURFL properties except "wurfl_id" are no longer injected by default
# -- and have to be explicitly specified.
#WurflRequestProperty wurfl_root_id
#WurflRequestProperty wurfl_isdevroot
#WurflRequestProperty wurfl_useragent
#WurflRequestProperty wurfl_engine_target
#WurflRequestProperty wurfl_useragent_priority
#WurflRequestProperty wurfl_info
#WurflRequestProperty wurfl_api_version
```

```
#WurflRequestProperty wurfl_last_load_time
#WurflRequestProperty wurfl_normalized_useragent
```

```
</IfModule>
```

Please refer to the directives guide that explains each element in detail (Table 1), their parameters, constraints, and default recommended settings.

**Table 1:**

Syntax	Description	Availability
<b>WurflRoot</b> <string>	This defines the location (path) of the WURFL data file.	1.4
<b>WurflUpdater</b> <string> <string>	<p>Allows seamless update of WURFL engine with new data downloaded from Scientiamobile. It takes two parameters:</p> <ul style="list-style-type: none"> <li>• the data url (taken from your personal Scientiamobile Vault account, choosing between two data file types: .zip or .xml.gz) Take care that WurflRoot file type and WurflUpdater data url file types match so you may need to change the WurflRoot file type accordingly.</li> <li>• the updater checking frequency (how often the updater checks for any new WURFL data file to be downloaded and used by the engine) which you can choose between DAILY and WEEKLY. In order to let the Updater perform its activities both the WurflRoot folder and file must be writable by Apache process. The wurfl-updater.log file in WurflRoot folder will contains details on Updater activity.</li> </ul>	1.8.3.1
<b>WurflPatch</b> <string>	This function to add one or more custom patch files to the WURFL repository.	1.4

Syntax	Description	Availability
<b>WurflCacheNull</b> or <b>WurflCacheLRU</b> <num>	In order to increase performance while processing real HTTP traffic, we suggest setting up a LRU cache. The LRU caching strategy will speed up lookup operations on User Agents that have already been processed by keeping them in a Least Recently Used map. By default the cache will be set to 30000 entries which accounts for 7 to 10 MB of additional memory usage. Specific concerns regarding memory usage apart, users are advised to size their cache generously (100,000 or more) to increase performance. For more information, please see <a href="#">LRU Cache Mechanism</a> .	1.4
<b>WurflRequestCapability</b> <string>	This function defines one or more WURFL Capabilities and/or WURFL Virtual Capabilities to be loaded in the memory run-time. Those WURFL capabilities/virtual capabilities will be loaded to Environment Variable for every HTTP requests.	1.4
<b>WurflRequestProperty</b> <string>	Enables injection of WURFL Properties (see section WURFL Properties below) in the Environment Variable for every HTTP requests.	1.8.2.0
<b>WurflUserAgentPriorityOverrideSideloadedBrowserUserAgent</b> or <b>WurflUserAgentPriorityUsePlainUserAgent</b>	You can choose between these two options in order to decide the useragent priority to use. <code>WurflUserAgentPriorityOverrideSideloadedBrowserUserAgent</code> tells WURFL to use the sideloaded browser user agent for device detection, while <code>WurflUserAgentPriorityUsePlainUserAgent</code> tells WURFL to use the plain user agent instead.	1.5.2

## Capabilities and Virtual Capabilities in WURFL Apache module

You can insert any number of `WurflRequestCapability` directives, specifying both static and virtual capabilities. Please see the following link for a [full list of all WURFL Capabilities](#). As of release 1.8.0.0, there is no need to include the mandatory capabilities list in the config file.

**Warning:** If you do not include any `WurflRequestCapability` directives in the config file, the Apache Module will load **ALL** WURFL capabilities found in the WURFL database. This scenario may be useful when applied with a pre-filtered version of the WURFL database with limited sets of capabilities. However, we strongly recommend using the predefined list of capabilities stored in the config file. Please be aware that loading too many capabilities (e.g. over 500 capabilities) may cause missing environment variables due to overflow limits and performance level may be degraded.

## WURFL Properties

The WURFL Apache module sets some convenient variables to retrieve information regarding the active WURFL configuration.

These variables are automatically calculated, and injected, into HTTP requests if specified in a `WurflRequestProperty` command.

Please note that the `wurfl_id` variable is injected by default so you don't have to specify it in the command.

### WURFL Properties Table

Variable Name	Contents	Availability
<code>wurfl_id</code>	Contains the device ID of the matched device. It is injected by default so you don't have to specify it in a <code>wurfl_request_property</code> command	1.4
<code>wurfl_root_id</code>	Contains the device root ID of the matched device.	1.4
<code>wurfl_isdevroot</code>	Tells if the matched device is a root device. Possible values are "TRUE" or "FALSE"	1.4
<code>wurfl_useragent</code>	The original useragent coming with this particular web request	1.5.1
<code>wurfl_api_version</code>	Contains a string representing the currently used Libwurfl API version	1.5.1
<code>wurfl_engine_target</code>	Contains a string representing the currently set WURFL Engine Target. Possible values are "DEFAULT", "FAST_DESKTOP_BROWSER_MATCH" or "INVALID"	1.5.1
<code>wurfl_info</code>	A string containing information on the parsed <code>wurfl.xml</code> and its full path	1.5.1

Variable Name	Contents	Availability
wurfl_last_load_time	Contains the UNIX timestamp of the last time WURFL has been loaded successfully.	1.5.1
wurfl_normalized_useragent	The normalized useragent.	1.5.1.3
wurfl_useragent_priority	The user agent priority used by WURFL.	1.5.2

## Testing WURFL Apache Module

For every HTTP request, the WURFL Apache Module will detect and push the device capabilities data to the Apache Environment. The WURFL variable names will be shown in uppercase and prefixed with WURFL\_. For example, the brand\_name capability will be shown as WURFL\_BRAND\_NAME.

To view WURFL capabilities' results, see the outputs of Apache Environment variables for each HTTP Requests. For example, use the following command in a PHP Script to show output of Apache Environment variables:

```
var_export($_SERVER);
```

Instead of PHP, you can run the simple CGI script called printenv.pl that usually comes along with the Apache installs.

Load the following CGI script to Apache environment:

```
#!/usr/bin/perl
print "Content-type: text/plain\n\nENVIRONMENT VARIABLES!\n\n";
print "$_ = '$ENV{$_}'\n" for sort keys %ENV;
```

### ***IMPORTANT - Decommissioning of MatchMode options***

*Prior to version 1.9 of the API, users could choose between Performance and Accuracy engine optimization options. These options had been introduced years ago to manage the behavior of certain web browsers and their tendency to present "always different" User-Agent strings that would baffle strategies to cache similar WURFL queries in memory.*

*As the problem has been solved by browser vendors, the need to adopt this strategy has diminished and ultimately disappeared (i.e. there was no longer much to be gained with the performance mode in most circumstances) and ScientiaMobile elected to "remove" this option to simplify configuration and go in the direction of uniform API behavior in different contexts.*

*When we wrote "remove" in the previous sentence, we were not being totally accurate. Customers who may find themselves in the unlikely situation of having to analyze significant amounts of legacy web traffic, may still enable the old high-performance internal behavior by enabling the ENGINE\_TARGET\_FAST\_DESKTOP\_BROWSER\_MATCH option in their engine target configuration. Please note that users with the old HIGH PERFORMANCE target engine will not receive an error. The old behavior will not be triggered, though. The default target (corresponding to the old High Accuracy) will be used instead.*

© 2017 ScientiaMobile Inc.

All Rights Reserved.

**NOTICE:** All information contained herein is, and remains the property of ScientiaMobile Incorporated and its suppliers, if

any. The intellectual and technical concepts contained herein are proprietary to ScientiaMobile Incorporated and its suppliers and may be covered by U.S. and Foreign Patents, patents in process, and are protected by trade secret or copyright law. Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from ScientiaMobile Incorporated.