



# Oracle WebCenter Sites: Mobility Server

Version 11.1.1

## Installation Guide

February 2012

# Table of Contents

[Table of Contents](#)

[Introduction](#)

[Pre-requisites](#)

[CMS](#)

[Database](#)

[Domain Names](#)

[Pre-requisite packages](#)

[Apache Configuration](#)

[Mobility Server Installation](#)

[Installation](#)

[Mobility Server Configuration](#)

[Configuring Database Connectivity](#)

[Configuring Application Settings](#)

[Start Memcached and Apache](#)

[Database Population](#)

[Content Server Registration](#)

[Device Database Initialization](#)

[Installation Verification Test](#)

[Appendix 1:](#)

[Cache expiration parameters in Editorial and Delivery environments](#)

[Appendix 2:](#)

[Installation of Publishing Updater performance software \(NetomatCacheUpdater\)](#)

[Appendix 3:](#)

[Multiple Instances of Mobility Server](#)

# Introduction

This document contains instructions for a standard installation of Mobility Server configured to use either a MySQL or Oracle 11G database.

[Appendix 1: Cache expiration parameters in Editorial and Delivery environments](#) contains information for configuring Mobility Server for better performance in delivery environments and [Appendix 2: Installation of Publishing Updater performance software \(NetomatCacheUpdater\)](#) contains information on configuring your environments for publishing.

This guide may be used to install multiple instances of Mobility Server on a single server. For more information, please see [Appendix 3: Multiple Instances of Mobility Server](#).

Please contact your account representative for help determining the best installation procedure for your organization's needs.

# Pre-requisites

## CMS

---

Mobility Server requires Content Server version 7.6.1 or greater w/ WEM. Note that while WEM is an optional component during the Content Server installation process, it is required for Mobility Server.

Mobility Server requires a user account to read and write content to and from Content Server. While an existing administrator account may be used, it is recommended that a new user is created with adequate roles and permissions.

## Database

---

Two schemas must be created for each non-delivery environment: one for MSAdmin, and one for MobilityServer. For delivery environments, only a MobilityServer database is necessary.

Privileges should be the same as those granted to the user for the Content Server data source.

Oracle 10G, Oracle 11G, and MySQL v5.0+ are supported.

Installations using Oracle databases require installation of the Oracle Call Interface. Use of DRCP is strongly recommended. Please contact your account representative for information regarding optimizing performance of Mobility Server w/ Oracle 11G using DRCP.

## Domain Names

---

**Mobility Server Unique Domain Name:** It is imperative that a unique domain name be created for each Mobility Server installation.

---

MSAdmin Unique Domain Name: A unique domain name is also required for each installation of the MSAdmin WEM application. Additionally, the MSAdmin domain name must be nested as a subdomain of the Mobility Server domain; for example, 'ms1.yourdomain.com' for Mobility Server and 'admin.ms1.yourdomain.com' for MSAdmin.

Later in this document the following example domain names will be referenced. Please replace them with your domain names:

**ms1.yourdomain.com**  
**admin.ms1.yourdomain.com**

## Pre-requisite packages

---

Depending on the initial configuration of your server, many of the required packages may already be installed or be available through your distribution's software repository. For certain server configurations, the following command will install all of the packages necessary for Mobility Server (in this case, with a MySQL back-end):

```
yum install httpd memcached mysql mysql-server php php-devel php-mysql  
php-pdo php-pear php-pear-HTTP-Request2 php-pecl-memcache php-tidy php-xml  
tidy zlib-devel
```

If any of these packages are not available, you may need to find them via another repository or manually install them. The following are the required software packages:

- Apache 2.2+
- PHP 5.3+
- memcached 1.4.5+
- PHP Libraries
  - php-mysql (*for MySQL-backed installations only*)
  - InstantClient/OCI (*for Oracle database-backed installations only*)
  - HTTP\_Request2 0.5.2+
  - php-memcache 3.0.5+
  - php-tidy 5.3+

- php-xml 1.15+
- ionCube loader (*provided*)

If the libraries listed above are installed through a package manager, each should be automatically configured. IonCube, which is provided with your installation package, can be configured by adding the following line to the end of the **php.ini** configuration file:

```
zend_extension = /<ioncube path>/ioncube_loader_lin_5.3.so
```

Be sure to select the appropriate ionCube library for your server architecture (32 or 64-bit).

## Apache Configuration

---

Apache should be configured with two virtual hosts -- one for MSAdmin and one for Mobility Server.

The following sample configuration can serve as a starting point. Remember that a VirtualHost stanza must be created for both MSAdmin and MobilityServer:

```
# MobilityServer Virtual Host
<VirtualHost *:80>
    ServerName ms1.yourdomain.com
    DocumentRoot /var/www/html/mobilityserver/web/
    DirectoryIndex index.php

    <Directory "/var/www/html/mobilityserver/web">
        AllowOverride All
        Allow from All
    </Directory>
    Alias /sf /var/www/html/mobilityserver/lib/vendor/symfony/data/web/sf
    <Directory
"/var/www/html/mobilityserver/lib/vendor/symfony/data/web/sf">
        AllowOverride All
        Allow from All
    </Directory>
</VirtualHost>
```

**Security Notice 1:** It is important to note that only the two Directory values above should be configured. The DocumentRoot must be set to the */web/* folder, **not** set to the Mobility Server root folder, to prevent configuration files from being exposed.

---

# Mobility Server Installation

## Installation

---

Once the Mobility Server manual installation package is extracted a 'mobilityserver' folder should exist. This folder contains a simple script named ***install.sh*** that must be run in a bash environment. This can be accomplished by simply running the following from within the new folder:

**bash install.sh**

After agreeing to the license, the folder will contain the following:

```
.
├── agreement.txt
├── install.sh
├── ioncube_32
├── ioncube_64
├── licenses
├── mobilityserver
├── mobilityserver.tar.gz
├── msadmin
├── msadmin.tar.gz
└── NetomatCacheUpdater.jar
```

The mobilityserver and msadmin folders should be copied to a directory of your choice. The following are common choices:

- /var/www/
- /opt/MS/

If the installation and configuration of the prerequisite software from the previous chapter has been completed, you can now move on to configuring Mobility Server in the next chapter.

# Mobility Server Configuration

## Configuring Database Connectivity

---

The "prod" section in both `mobilityserver/config/databases.yml` and `msadmin/config/databases.yml` should be edited to match your database configuration. The only difference between the two files should be the **dsn**, **username**, and **password** fields as MSAdmin and MobilityServer require different schemas.

Oracle 11G Example:

```
prod:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn: 'oracle:dbname=<HOST>/<SID>;charset=AL32UTF8;persistent=false'

      username: <oracle username>
      password: <oracle password>
      encoding: utf8
      attributes:
        use_native_enum: true
        quote_identifier: true
```

\*Note that use of DRCP is recommended.

MySQL Example:

```
prod:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn: 'mysql:host=localhost;dbname=mobilityserver'
      username: <mysql username>
      password: <mysql password>
      encoding: utf8
      attributes:
        use_native_enum: true
        quote_identifier: true
```



**Security Notice 2:** Only the directories listed in the [Apache Configuration](#) section should be configured to be served by Apache. Care should be taken not to accidentally expose the root Mobility Server as a DocumentRoot to Apache, and permissions for this configuration folder should be set so that only the user running the Apache process can read from it. Once installation is completed, it is recommended that the database credentials configured through this file be limited to only reading and writing to and from the existing tables in the Mobility Server database.

## Configuring Application Settings

---

**mobilityserver/apps/frontend/config/app.yml** and **msadmin/apps/netowem/config/app.yml** should be configured to match your environment.

These configuration files contain up-to-date comments describing the directives contained within them. The following options are critical to begin using Mobility Server, though they may not each exist in both configuration files. Please refer to the sample configuration file provided with your installation package

allowed_manager_ips	A comma-delimited list of IPs that are allowed to access administrative functions. Please see the configuration file for examples.
oraclecms_server_host	The hostname or IP address of the WebCenter Sites or FatWire Content Server
oraclecms_server_port	The port of the WebCenter Sites or FatWire Content Server
oraclecms_user	The username for the user described in the <a href="#">CMS</a> prerequisite section of this document
oraclecms_password	The password for the user described in the <a href="#">CMS</a> prerequisite section of this document
oraclecms_site_name	The site selected for this Mobility Server installation. See the Implementation Guide for details.
netowem_server_host	The hostname of MSAdmin as described in the <a href="#">Domain Names</a> section of this document
netowem_server_port	The port MSAdmin is configured to use
ms_server_host	The hostname of Mobility Server as described in the <a href="#">Domain Names</a> section of this document
ms_server_port	The port Mobility Server is configured to use

The various caching and performance options are detailed in [Appendix 1: Cache expiration parameters in Editorial and Delivery environments](#).

**Security Notice 3:** Only the directories listed in the [Apache Configuration](#) section should be configured to be served by Apache. Care should be taken not to accidentally expose the root Mobility Server as a DocumentRoot to Apache, and permissions for this application configuration folder should be set so that only the user running the Apache process can read from it.

## Start Memcached and Apache

---

Before proceeding, ensure that Memcached, Apache and your database server have been started and are running properly. Ensure that the ports they are listening on match the configuration settings you provided in the [Configuring Application Settings](#) section of this document, and confirm connectivity to Content Server and your database server.

## Database Population

---

The following command will build the schemas and populate them with base data for Mobility Server and MSAdmin, respectively (after prompting for confirmation):

```
cd /<mobility server path>/mobilityserver  
php symfony doctrine:build --all --and-load --env=prod
```

```
cd /<msadmin path>/msadmin  
php symfony doctrine:build --all --and-load --env=prod
```

Please see [Security Notice 2](#) for more information on changing database permissions once installation has been completed.

## Content Server Registration

---

The following commands will configure MSAdmin and Mobility Server to work with Content Server. The first task below will also install the MSAdmin WEM application.

```
cd /<msadmin server path>/msadmin
php symfony mobilityserver:configure --env=prod
php symfony mobilityserver:createapplication --env=prod

cd /<mobility server path>/mobilityserver
php symfony mobilityserver:configure --env=prod
```

## Device Database Initialization

---

The following command must be run to initialize the device database:

```
cd /<mobility server path>/mobilityserver
php symfony mobilityserver:parsedevicedb --env=prod
```

## Installation Verification Test

---

Once Apache, memcached and your database server are running, you can run a command-line task to ensure that Mobility Server has been properly configured:

```
cd /<mobilityserver path>/mobilityserver
php symfony mobilityserver:testinstall --env=prod --host=ms1.yourdomain.com:80
```

At this point you may discover that Mobility Server has insufficient permissions to certain folders. Please adjust permissions to these folders accordingly, only permitting the Apache user with as much access as is required, and only for the directories indicated. Please confirm that all permissions for your installation are configured properly and in accordance with your organization's security policy.

Once this test succeeds, logging into Content Server should now present you with an option for MSAdmin, and clicking through to it will allow you to preview a landing page for your future mobile site. For more information on building your first mobile site, please see the [Mobility Server Site Implementation Guide](#).

## Appendix 1:

# Cache expiration parameters in Editorial and Delivery environments

The following properties are used to configure cache expiration in app.yml. All expiration times are in seconds.

**memcached\_frontend\_cache\_lifetime** - expiration time of the frontend memcache, which caches pages on a per-user-agent basis. In Editorial environments, we recommend using a value of 0 (no caching). In Delivery environments, we suggest a fairly long value such as 3600 (one hour).

**properties\_cache\_lifetime** - expiration time of the properties cache. In Editorial environments, we recommend using a value of 0 (no caching). In Delivery environments, we suggest a long value such as 86400 (one day).

**rest\_call\_cache\_lifetime** - expiration time of the REST cache. In Editorial environments, we recommend using a value of 5 (very short). In Delivery environments, we suggest a long value such as 86400 (one day) because REST results should be removed from the cache by the Publishing process.

**geolocation\_cache\_lifetime** - expiration time of the Geolocation cache. In both Editorial and Delivery environments, we suggest a long value such as 86400 (one day) or more, since this data should rarely change.

**query\_cache\_lifetime** - expiration time of the Query cache. In both Editorial and Delivery environments, we suggest a long value such as 7200 (2 hours).

**result\_cache\_lifetime** - expiration time of the Query Results cache. In both Editorial and Delivery

environments, we suggest a long value such as 7200 (2 hours).

**syndication\_data\_ttl** - expiration time of the syndication data cache, which stores data from syndicated feed calls such as RSS or Atom. In both Editorial and Delivery environments, we suggest a short value such as 30 seconds, though this may be varied based on the characteristics of the feeds used in a specific environment.

**use\_appcache** - takes the values true or false. True means use external storage for the appcache, and is recommended for delivery environments. False is recommended for Editorial. Please see the Site Implementation Guide for more information in configuring offline storage using ApplicationCache.

## Appendix 2:

# Installation of Publishing Updater performance software (NetomatCacheUpdater)

Mobility Server content can be more efficiently updated by use of the NetomatCacheUpdater software.

To install and configure NetomatCacheUpdater two files need to be put in the Content Server (cs) webapp's directories on a target (publishing destination) Content Server:

- 1) NetomatCacheUpdater.jar should be put in the directory **cs/WEB-INF/lib**
- 2) netomatCacheUpdater.properties should be put in **cs/WEB-INF/classes**

To configure NetomatCacheUpdater, edit the netomatCacheUpdater.properties file, replacing "[mobilityServer]" with the actual mobility server name. If there's more than one, you can use a list of server URLs separated by commas (no spaces).

Also on the destination ContentServer, edit the file

`$TOMCAT_HOME/webapps/cs/WEB-INF/classes/AdvPub.xml` and replace the line:

```
<bean id="PageCacheUpdater" class="com.fatwire.realtime.PageCacheUpdaterImpl"
singleton="false">
```

with this line:

```
<bean id="PageCacheUpdater" class="net.netomat.fwcache.NetomatCacheUpdater"
singleton="false">
```

After installing and configuring NetomatCacheUpdater, restart the application server.

If there are any issues please send your support representative the application server log file and also the Content Server "futuretense.txt" log file.



## Appendix 3:

# Multiple Instances of Mobility Server

Multiple instances of Mobility Server may be installed on a single server. This is a common configuration for organizations that want to deploy mobile sites for more than one of their Content Server sites. To accomplish this, an additional pair of databases (please see the [Database](#) section) and an additional pair of domain names (please see the [Domain Names](#) section) should be created.

Although it is possible to install multiple copies of the [Pre-requisite packages](#), this is not a requirement and rarely necessary. Instead, repeat just the [Apache Configuration](#) section using the new domain names from the [Pre-requisites](#) chapter.

Once that is complete, the [Mobility Server Installation](#) chapter can be repeated, substituting in new paths for the additional copy of the Mobility Server software (the mobilityserver and msadmin directories). Likewise, the [Mobility Server Configuration](#) steps can be repeated as well, provided you begin in the newly created folders where appropriate and substitute in the appropriate domain names and database settings for the new installation.