Unified Notification Service

Installation manual

Prepared by

Nedeljko Pavlovic, European Dynamics

October 2006

Version 1.1

***Administrator manual***

Version management and approval

|  |  |  |  |
| --- | --- | --- | --- |
| No | Date | Changes | Author |
| 1.0 | 10/10/06 | Release to EEA | Nedeljko Pavlovic |
| 1.1 | 20/20/06 | Conversion to ODT | Søren Roug |

*Project manager: Dr. Stelios Papastratos*

Contents

1. Introduction [4](#_toc118)

2. Software requirements [4](#_toc120)

3. Installing software tools [4](#_toc135)

3.1. Installing Java tools [4](#_toc136)

3.2. Installing MySQL Database Server [5](#_toc156)

3.3. Installing Apache Tomcat [5](#_toc158)

3.4. Installing Apache HTTP server (Optional) [6](#_toc161)

3.5. Installing Python tools [6](#_toc163)

4. Installing and configuring the UNS [7](#_toc179)

4.1. Obtaining Unified Notification Service [7](#_toc180)

4.2. Creating UNS database schema [7](#_toc211)

4.3. Configuring and building UNS [8](#_toc227)

4.4. Configuring Tomcat [11](#_toc317)

4.5. Configuring Apache HTTP Server (Optional) [12](#_toc333)

4.6. Deploying UNS [13](#_toc356)

4.7. Controlling the UNS daemon [14](#_toc373)

Document conventions

Acronyms/Abbreviations:

EEA - European Environment Agency

JSP - Java Server Pages

EIONET - The European Environmental Information and Observation

Network

HTTP - Hypertext Transfer Protocol

RDF - Resource Description Framework

RSS - Really Simple Syndication

XML - Extensible Mark-up Language

XML-RPC - Remote procedure calling using HTTP as the transport protocol

and XML as the encoding protocol.

UNS - Unified Notification Service

# Introduction

This document provides instructions for the installation and configuration of the software required for the Unified Notification project.

# Software requirements

The Unified Notification Service has been tested to run on the Linux platform. It has been tested on RedHat Enterprise, Gentoo and Slackware distributions.

Given that the UNS has been developed as standard J2EE application, there shouldn’t be any limitations for running on other platforms like Windows and Solaris.

The REPORTNET UNS relies on the following software components:

* Java Runtime Environment 1.5.0\_x
* MySQL Database Server versions 4.1.8 or greater
* Apache Tomcat 5.5.15
* Jython 2.2
* Apache HTTP server (Optional)
* Apache Ant 1.6.5 (Optional)
* Python 2.3.5
* mysql-python 1.2.0
* python-ldap 2.0.10
* libxml2-python
* UNS application source

# Installing software tools

## Installing Java tools

### Installing Java runtime environment

The Unified Notification Service has been implemented by using Java programming language, therefore it is mandatory for running server box to have installed J2SE[[1]](#footnote-2).

The UNS requires latest stable version from the 1.5.0 branch.

It may be downloaded from at <http://java.sun.com/javase/downloads/index.jsp> . Full and comprehensive installation documentation is available for download at  [http://java.sun.com/j2se/1.5.0/install.html .](http://java.sun.com/j2se/1.5.0/install.html)

Please note that you need to download and install J2SE SDK in case that you want to build UNS from the source.

Setting up Java run time environment for the UNS is explained at section 4.4.2.

### Installing Apache Ant

The Unified Notification Service is using Apache Ant in order to make build process easy. Apache Ant is a Java-based build tool.

Apache Ant can be downloaded from the [Apache Ant download site](http://ant.apache.org/bindownload.cgi).

In order for Apache Ant to be installed, all you need to do is to extract the archive in your local system and then set ANT\_HOME in according to your environment and add ANT\_HOME/bin in your system PATH.

You also need to define the JAVA\_HOME environment variable. This variable should be the directory where a Java Development Kit is installed (note that a JRE is not sufficient)

### Installing Jython 2.2

Download latest release of the Jython from the [Jython sourceforge dowload site](http://switch.dl.sourceforge.net/sourceforge/jython/jython_Release_2_2alpha1.jar) .

If you have GUI environment execute command <java interpreter> -jar jython\_Release\_2\_2alpha1.jar

If you do not have a GUI, then add -o dir\_to\_install\_to to the command. Jython will install to the specified directory without bringing up the graphical installer. E.g. to install all modules to a Jython-2.1 subdirectory in the current directory do:

<java interpreter> -jar jython\_Release\_2\_2alpha1.jar -o Jython-2.1 demo lib source

You will be prompted for simple information regarding jython installation.

To check out successful installation access the jython instillation directory and run jython bash script which will invoke jython interpreter.

You may have to change file execution mode with command "chmod +x jython"

## Installing MySQL Database Server

You may download the MySQL installation from <http://dev.mysql.com/downloads/mysql/4.1.html>. The download and installation of the latest stable release from the 4.1.x branch that the MySQL website advices is highly recommended. Currently it is version 4.1.12. A comprehensive and full installation guide of MySQL can be found at <http://dev.mysql.com/doc/mysql/en/installing.html>

## Installing Apache Tomcat

Apache Tomcat is the servlet container that is used in the official Reference Implementation for the [Java Servlet](http://java.sun.com/products/servlets) and [JavaServer Pages](http://java.sun.com/products/jsp) technologies. Since the UNS has been developed by using those technologies, it is recommended to run under Apache Tomcat. Apache Tomcat can be downloaded from the [Apache Tomcat download site](http://jakarta.apache.org/site/downloads/downloads_tomcat-5.cgi).

The UNS requires latest stable release from the 5.0.x branch

## Installing Apache HTTP server (Optional)

Apache can be downloaded from the [Apache HTTP Server download site](http://httpd.apache.org/download.cgi) which lists several mirrors. The build process of apache can be found at http://httpd.apache.org/docs/1.3/ and it allows the user to customize the HTTP server to suit his/her needs.

## Installing Python tools

### Installing Python

The Unifed Notification Service has been implemented by using the Python programming language, therefore it is mandatory that Python must be installed on the server that UNS will be running. The UNS requires Python 2.3.5 or later. Python 2.3.5 may be downloaded from <http://python.org/2.3.5>.

### Installing MySQL for Python

The UNS uses the MySQL database as the main persistence storage. Therefore the interface to the MySQL database server that is provided by the Python database API is required.

This interface may be downloaded from <http://sourceforge.net/projects/mysql-python>

A comprehensive and full installation guide for the MySQL-Python interface can be found at <http://sourceforge.net/docman/display_doc.php?docid=26237&group_id=22307>

### Installing python-ldap

UNS uses the EEA LDAP server for authentication purposes as well as to retrieve user’s and EEA roles information. Communication with the LDAP has been implemented by using the LDAP client API for Python (python-ldap). The UNS has been tested to run with python-ldap versions 2.0.4-2.0.10. Version 2.0.10 is recommended.

It may be downloaded from <http://python-ldap.sourceforge.net/download.shtml>

### Installing libxml2-python

UNS is using stylesheets to transform RSS feeds into the RDF form. Transformations are handled by the libxml2-python API.

It can be downloaded from [http://xmlsoft.org](http://xmlsoft.org/)

\* **Important note**: The libxml2-python depends on the libxml2 and libxslt packages.

It shall be ensured that libxml2 and libxsl are properly installed. These two packages become standard and therefore they should be included in a typical Linux/Unix distribution; however it is recommended that these packages are installed. Also please note that the version of the libxml2-python shall comply with the OS libxml2 and libxslt versions that have been have installed. The best is to install the version shipped with the OS installation (if exists).

# Installing and configuring the UNS

## Obtaining Unified Notification Service

To get latest version of the Unified Notification Service source you need to check it out from the EIONET’s Subversion repository. To checkout source just use the following command on the command prompt:

shell> svn checkout http://svn.eionet.eu.int/repositories/Reportnet/UNS/trunk

After execution of the above command UNS source will be downloaded. The UNS has common directory layout as suggested by Maven foundation in order to allow users familiar with one Maven project to immediately feel at home in UNS project.

The UNS directory layout that will appear after downloading source is as follows:

**/trunk**

**+- src/**

**| +- main/**

**| | +- java/** *UNS java source code*

**| | | +- ...**

**| | +- python/** *Python* *source code*

**| | | +- ...**

**| | +- resources/** *Configuration files*

**| | | +- ...**

**| | +- public\_html/**  *Web* *files (JSP, Images, CSS, Java scripts)*

**| | | +- ...**

**| | +- sql/**

| | | **+- UNS2\_create.sql** *Script for creating UNS database schema*

**| +- test/**

**| | +- java/**

**| | | +- ...**

**| | +- resources/**

**| | +- ...**

**+- project.xml**

**+- project.properites**

**+- README.txt**

**+- LICENSE.txt**

**+- build.xml**

**+- default.properties**

## Creating UNS database schema

The UNS database structure as well as inital data is stored inside create\_UNS.sql file. Location of the file in the source distribution is outlined in section 3.5.4.

To import the database structure, use the MySQL client connecting as DB administrator and follow this command on the system shell:

shell> mysql -u root -p <src/main/sql/UNS2-create.sql

After executing this command the Unified Notification database is created.

We strongly recommend to create user who will be allowed to perform only basic operations on the UNS schema such as reading, inserting, updating, deleting tables/rows and use the DB administrative user for running the UNS.

Therefore the next step is to create a UNS user. To create the UNS application user and grant him appropriate privileges the following commands must be executed:

* Grant privileges:

mysql> grant select,insert,update,delete,drop on uns.\* to uns@localhost identified by 'your\_password';

The value for the Host part after the '@' should be ’localhost’ if the UNS application and MySQL will be running on same machine. If MySQL database and the UNS application are running on different servers then you need to provide the address of the MySQL host e.g. ’UNS.eionet.europa.eu’.

* Flushing privileges:

mysql> FLUSH PRIVILEGES;

The reason for using FLUSH PRIVILEGES when you create accounts with INSERT is to tell the server to re-read the grant tables. Otherwise, the changes go unnoticed until you restart the server.

UNS schema and application user shall be created now and you may exit from the mysql client by using the following command:

mysql> quit;

\* **Important note**: Initially UNS (Unified Notification Service) channel will be created. If you want to allow the UNS to send notifications to your UNS installation, you must not delete this channel

## Configuring and building UNS

UNS has two installation directories. One containing the Tomcat webpages, and another one for the configuration files, acl files, log files etc. The first directory is the location you place the uns.war file in. The second directory is called the UNS\_HOME below.

### Setting up UNS configuration parameters

#### Ant build script configuration

The following parameters need to be set in default.properites file. This file is located in the root directory of the UNS source distribution.

uns.home

this is the home directory of the UNS installation where configuration and logging files are stored. Ant will create this directory in the "prepare" target. The uns-config.xml configuration file is located in this folder and need to be configured manually.

This directory will be referenced in the following sections as the UNS\_HOME.

app.name

Name of the application. This name will be used as name for Web application (WAR) archive file. If you would like to install the UNS into root application context this name must be ROOT.

context.path

Context path to which this application should be deployed

catalina.home

The directory in which you have installed a binary distribution of Tomcat 5. This will be used by the "compile" target.

ldap.url

The URL of the EIONET ldap service as is used in the EIONET directory service library.

db.user

MySQL dB username for dB connections

db.password

MySQL dB password for dB connections

db.url

The JDBC connection url for connecting to your MySQL dB. You shouldn’t specify additional parameters here. The default arguments added to the connection url are “useUnicode=true” and “characterEncoding=UTF-8”. If you want to change these parameters please edit /src/main/public\_html/META-INF/context.xml before building process

uns\_python\_source.home

the home directory of the UNS python source code It will be created by the ant "prepare" target The UNS python source including daemons will be stored here. This directory will be referenced in the following sections as the UNS\_PYTHON\_SOURCE\_HOME.

python.home

Location of the python interpreter

jython.home

The home directory of the your jython installation

uns.url

The web address of the UNS application. This link will be sent together with notifications for easy unsubscribing service.

#### Configuring parameters in the uns-config file

It will be necessary to set up appropriate parameters related to database, jabber, ldap and mail service for your environment in the uns-config.xml file. You can find this file in the src/main/resources/ folder in the source distribution. In the building process after you perform ant “prepare” target this file will be copied into UNS\_HOME folder. Both UNS java application and python daemons read parameters from that location. Also UNS application provides configuration of those parameters through web interface and any changes on parameters during runtime will be automatically reflected into UNS\_HOME/uns-config.xml file.

### Security configuration

#### Configuring users authentication and access control

The UNS groups and permissions are already defined. The UNS is using two groups: admin and xmlrpc. You need to assign to admin group any users who will be able to perform UNS administrative operations. Users who will be able to use the UNS XML-RPC features shall be assigned to the xmlrpc group. Because UNS now uses CAS authentication service local users are unable to create RPC channels via web interface. Only users from EIONET ldap service that belong tho xmlrpc group may create a rpc channel via web interfece. Both types of users (local and ldap) may create and push events trought XML-RPC api.

To assign users to desired groups you need to edit UNS\_HOME/acl/UNS.group file as the following:

admin:auser\_1,auser\_2,…

xmlrpc:user\_1,user\_2,…

\* Important note: You shall not delete or rename any of those ACL files.

#### Configuring CAS SSO service

The CAS client is implemented in the class EionetCASFilter. EionetCASFilter extends standard java CAS filter so all configuration parameters from standard java CAS filter are supported. For detailed description of the parameters meaning please see: <http://www.ja-sig.org/wiki/display/CAS/Using+CASFilter>

CAS service may be easy configured by changing few configuration parameters for the EionetCASFilter in the web.xml file. For detailed description of the parameters meaning please see: <http://www.ja-sig.org/wiki/display/CAS/Using+CASFilter>. An additional parameter is represented “eionetLoginCookieDomain” which automatically authenticate user into applications if the user has established CAS SSO session.

This is an example how the EionetCASFilter filter should be configured.

<filter>

<filter-name>CAS Filter</filter-name>

<filter-class> com.eurodyn.uns.web.filters.EionetCASFilter</filter-class>

<init-param>

<param-name>edu.yale.its.tp.cas.client.filter.loginUrl</param-name>

<param-value>https://sso.eionet.europa.eu/login</param-value>

</init-param>

<init-param>

<param-name>edu.yale.its.tp.cas.client.filter.validateUrl</param-name>

<param-value>https://sso.eionet.europa.eu/serviceValidate</param-value>

</init-param>

<init-param>

<param-name>edu.yale.its.tp.cas.client.filter.serverName</param-name>

<param-value>uns.eionet.europa.eu</param-value>

</init-param>

<init-param>

<param-name>edu.yale.its.tp.cas.client.filter.wrapRequest</param-name>

<param-value>true</param-value>

</init-param>

<init-param>

<param-name>eionetLoginCookieDomain</param-name>

<param-value>eionet.europa.eu</param-value>

</init-param>

</filter>

You should also import tomcat certificate of your CAS server into default keystore of the trusted certificates, that is $JAVA\_HOME/bin/keytool -import -file server.crt -keypass changeit -keystore $JAVA\_HOME/jre/lib/security/cacerts.

### Building UNS

#### Source distribution

In order to build the Unified Notification service you should have installed J2SE SDK and Apache Ant as described in section 3.1. The build process is straight-forward. The following steps need to be performed:

* Clean all previous builds. This can be achieved by using the following command:

shell> ant clean

* Prepare basic setup. This task will create UNS\_HOME and UNS\_PYTHON\_SOURCE folders and replace appropriate tokens in the configuration files with values that you will provide in the default.properties file..

shell> ant prepare

* Build the UNS by executing the following command:

shell> ant compile

* Make war archive by executing the following command.

shell> ant dist

The UNS web archive will be ready inside dist directory.

## Configuring Tomcat

### Installing MySQL JDBC driver

The Unified Notification Service needs a Java driver for accessing MySQL database. The JDBC (Java DataBase Connection) driver for MySQL can be downloaded from

<http://www.mysql.com/downloads/api-jdbc-stable.html>. The UNS has been developed and tested with MySQL Connector/J version 3.1.8.

In order to install JDBC driver copy “mysql-connector-java-\*.jar” into the Tomcat’s CATALINA\_\_HOME/common/lib directory.

### Setting up Java Runtime Environment

In order to achieve an optimal application performance we recommend the following java runtime options:

* -Xms512M   
  Minimum heap size that will be allocated by the JVM.
* -Xmx512M   
  Maximum heap size that will be allocated by the JVM.
* -XX:+UseParallelGC   
  This option represents parallel garbage collecting. It will increase overall performance of the container. It shall be used ONLY on multi CPU machines.

These options need to be provided as JAVA\_OPTS environment variable in order to allow Apache Tomcat to use it e.g.

shell> export JAVA\_OPTS="–Xms512M -Xmx512M -XX:+UseParallelGC"

### Preparing Tomcat to work with HTTP server (Optional)

Tomcat must be enabled to use AJP13 protocol i.e. it should start listening for AJP13 requests. So an additional **<Connector>** element in *server.xml* must exist:

<Connector port="8009" enableLookups="false" redirectPort="8443" debug="0"

protocol="AJP/1.3" />

## Configuring Apache HTTP Server (Optional)

The suggested Web Server connector is Mod\_jk. Mod\_jk can be obtained in two formats binary and source form <http://www.apache.org/dist/jakarta/tomcat-connectors/jk/>. Depending on the platform you are running your web server on, a binary version of mod\_jk may be available. It is recommended to use the source version for better integration in production environment.

Build and install *jk-module* from the source*:*

shell> ./configure –with-apxs=/path/to/apache/install/dir/bin/apxs

shell> make

shell> make install

Create file *workers.properties* in the apache configuration directory and add the following lines:

worker.list=UNSWorker

worker. UNSWorker.port=8009

worker. UNSWorker.host=*TomcatServerName or IP*

worker. UNSWorker.type=ajp13

Add following lines into Apache's *httpd.conf* file:

JkWorkersFile */path/to/workers/config/file/*workers.properties

JkLogFile */path/to/connector/log/file/*mod\_jk.log

JkLogLevel debug

JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

JkRequestLogFormat "%w %V %T"

JkMount /\* UNSWorker

At the end, restart Apache web server:

shell> apachectl restart

## Deploying UNS

Actual deploy of the Unified Notification application can be performed by using many different approaches. This section will describe two most common ways of performing deployment under Apache Tomcat. Since the UNS will be deployed in uns context you must remove existing uns application before continuing with deployment.

You can just delete it e.g.:

shell> rm -R $CATALINA\_HOME/webapps/uns

### Deployment by copying WAR

The first and most common way is to simply place application WAR file under $CATALINA\_HOME/webapps directory.

* Copy WAR:

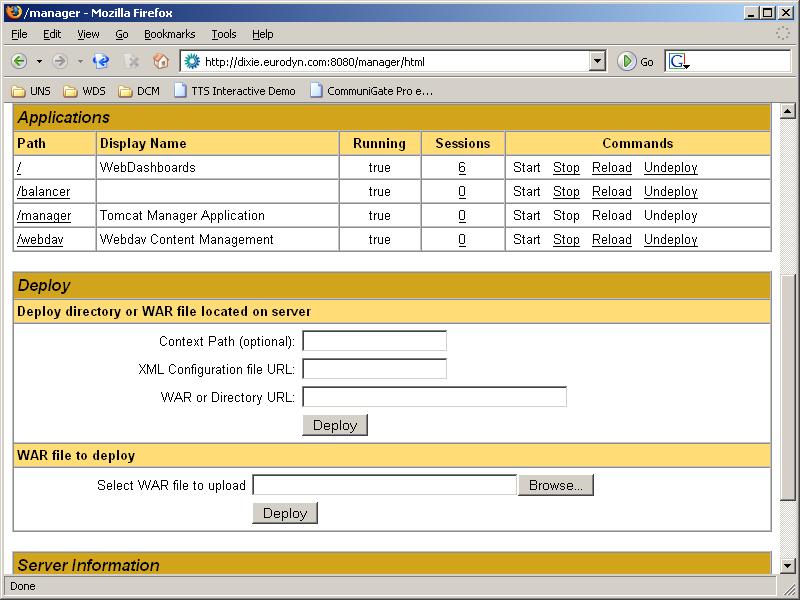
shell> cp dist/uns.war $CATALINA\_HOME/webapps

* Start Tomcat:

shell> $CATALINA\_HOME/bin/startup.sh

### Deployment by using Manager Application

The Second way to perform deployment is by using Apache Tomcat Manager Application. So user needs to login into the manager application and perform desired administrative operation which will be in this case deployment of the UNS.

Illustration 1: The Tomcat Manager application

- If the UNS WAR has been built on the production server where Tomcat is running then the administrator only needs to use “Deploy directory or WAR file located on server” option and provide full path to the web archive, e.g. file:/home/user/trunk/target/uns.war

- If the UNS WAR has been built on the local PC, the administrator only needs to provide under “WAR to deploy” section the local location of the UNS WAR. The manager application will perform upload and deployment.

\* **Important note**: These instructions assume that the WAR file is unpacked, which is the default is setting in a Tomcat installation. To check the setting, find the following lines in $CATALINA\_HOME/conf/server.xml:

<!-- Define the default virtual host -->   
<Host name="localhost" debug="0" appBase="webapps" unpackWARs="true">

## Controlling the UNS daemon

The UNS daemon runs in order to harvest new events, generate and send notifications to the users. It is controlled by using a shell script UNS\_PYTHON\_SOURCE/UNS/unsd.sh.

The UNS daemon achieves its function throw two threads. The first thread is called *Harvester* and is responsible for harvesting the events and the second is called *Notificator* and is responsible for generating and sending notifications to the users.

The following parameters are contained inside script UNS\_PYTHON\_SOURCE /UNS/unsd.sh.

* PYTHON - Location of the Python executable
* INSTANCE\_HOME - Location of the uns python source code
* UNS\_ROOT\_CONTEXT -WEB context root or the url address of the Your UNS
* UNS\_HOME - home UNS directory where is uns-config.xml file stored

Initial configuration of the script parameters is provided after And build process. You may also place uns python source on the other location in the system but you will be required to setup these parameters manually in the unsd.sh script.

After setting those parameters, it will be necessary to run the unsd.sh script.

The UNS script will inform the user with an appropriate message in case that wrong parameters have been set-up.

The control can be done in the following way:

* Start daemon

shell> ./unsd.sh start

* Stop daemon

shell> ./unsd.sh stop

* Restart daemon

shell> ./unsd.sh restart

Activities of the UNS daemon will be logged inside the UNS\_PYTHON\_SOURCE/log/unsd.log file for both approaches.

1. Java 2 Platform, Standard Edition <http://java.sun.com/j2se> [↑](#footnote-ref-2)