



NORDUGRID-MANUAL-12

20/7/2009

# LUNARC APPLICATION PORTAL

*Installation manual*

Jonas Lindemann\*

---

\*jonas.lindemann@lunarc.lu.se

# Contents

<b>1</b>	<b>Installation</b>	<b>2</b>
<b>2</b>	<b>Prerequisites</b>	<b>2</b>
<b>3</b>	<b>Installing WebWare for Python</b>	<b>2</b>
<b>4</b>	<b>Setting up a WebWare application instance</b>	<b>3</b>
<b>5</b>	<b>Configuring the Webware application instance</b>	<b>3</b>
<b>6</b>	<b>Configuring the application instance for LAP</b>	<b>4</b>
<b>7</b>	<b>Building the mod_webkit module for the Apache web server</b>	<b>4</b>
<b>8</b>	<b>Configuring the Apache web server</b>	<b>4</b>
<b>9</b>	<b>Installing the Lunarc Application Portal</b>	<b>5</b>
<b>10</b>	<b>Configuring the portal</b>	<b>5</b>
<b>11</b>	<b>Creating a user</b>	<b>7</b>

# 1 Installation

This document provides manual instructions for installing the portal. This can be useful when custom setups or different Linux distributions are used.

## 2 Prerequisites

This installation manual requires that ARC-1.x is built and configured with the python-bindings for `libarcclient`. This manual does not cover how to configure SELinux to work with Apache and the WebWare application server.

## 3 Installing WebWare for Python

The first step in installing the portal is to install and configure the WebWare for Python application server. Webware tarballs can be downloaded from:

<http://sourceforge.net/projects/webware/>

The tar-package can be downloaded and extracted with the following commands:

```
[root@grid sw]# wget [sourceforge url]
--13:57:29-- http://[...sourceforge...]/Webware-1.0.2.tar.gz
          => 'Webware-1.0.2.tar.gz'
.
.
.
100%[=====>] 936,493      164.42K/s   ETA 00:00

13:57:35 (168.79 KB/s) - 'Webware-1.0.2.tar.gz' saved ...
[root@grid sw]# tar xvzf Webware-1.0.2.tar.gz
Webware-1.0.2/
Webware-1.0.2/KidKit/
Webware-1.0.2/KidKit/Docs/
Webware-1.0.2/KidKit/Docs/RelNotes-1.0.2.phtml
.
.
```

To install Webware, the provided installation scripts must be run:

```
[root@grid sw]# cd Webware-1.0.2
[root@grid Webware-1.0.2]# python ./install.py
Webware for Python 1.0.2
Installer

    Cur Date: Tue Sep 25 14:08:58 2007
    Python: 2.3.4 (#1, May  2 2007, 19:26:00)
.
.
Choose a password for the WebKit Application Server.
If you will just press enter without entering anything,
a password will be automatically generated.
Password:
```

The installation script stops after a while to ask for a password. Press enter to generate a random password. As we are not going to use the WebWare administration interface a random password will suffice. If everything went well the following text is displayed indicating a successful installation:

```
Welcome to Webware!

You can already try out the WebKit application server.
Start it with "WebKit/AppServer" and point your browser
to "http://localhost:8080".

Browsable documentation is available in the Docs folders.
You can use "Docs/index.html" as the main entry point.

Installation is finished.
```

## 4 Setting up a WebWare application instance

Before the portal can be installed, a WebWare application instance has to be created. A WebWare application instance is a separate directory containing startup and configuration files for an application. The WebWare installation directory contains command line tools for accomplishing this. The following procedure will setup a application instance at /opt/lap with a context directory /opt/lap/context.

```
[root@grid Webware-0.9]# cd
[root@grid Webware-0.9]# cd bin
[root@grid bin]# ./MakeAppWorkDir.py -c context /opt/lap
Making a new WebKit runtime directory...

Creating the directory tree...
    /opt/lap/
    /opt/lap/Cache
.
.
.
```

## 5 Configuring the Webware application instance

To test the instance the application server can be started using the AppServer command:

```
[root@grid lap]# cd /opt/lap
[root@grid lap]# ./AppServer
Starting WebKit.ThreadedAppServer...

WebKit AppServer 1.0.2
.
.
.
Creating 10 threads.....
Listening for Adapter on 127.0.0.1:8090
Listening for HTTP on 127.0.0.1:8080
Ready (0.14 seconds after launch).
```

If everything has been setup correct, a web page should be displayed when a browser is directed to `http://localhost:8080`. WebWare comes with a built in web server that is configured to start by default. This server will be disabled in when deploying the application using the Apache web server.

## 6 Configuring the application instance for LAP

The Lunarc Application Portal requires the use of HTTPS for transfer of sensitive login data. As the built in Web server does not support this protocol it has to be disabled. This is done in the `[instance dir]/Configs/AppServer.config`. In the file the `EnableHTTP` configuration value must be set to `False`.

```
.  
.br/>EnableHTTP = False # enable built-in Webserver  
HTTTPort = 8080  
.br/>.
```

## 7 Building the mod\_webkit module for the Apache web server

To be able to use WebWare from the Apache Web Server the `mod_webkit` module has to be build. The source for this module is available in the WebWare-0.9.x source dir. To build the module, the `apxs` build tool is needed. On a RedHat-based distro it can be installed using `yum`

```
# yum install httpd-devel
```

Next the module is built and installed.

```
# cd {WEBWARE_DIR}/WebKit/Adapters/mod\_webkit2  
# make  
# make install
```

## 8 Configuring the Apache web server

To use the Apache Web server with the Lunarc Application Portal it has to be configured for the HTTPS protocol. To make access to the portal more secure, users will be authenticated using X509 certificate. A basic configuration file (`/etc/httpd/conf.d/lap.conf`) for a RedHat based distribution is shown below:

```
LoadModule webkit_module modules/mod_webkit.so  
  
<Location /lap>  
WKServer localhost 8086  
SetHandler webkit-handler  
SSLRequireSSL  
AuthName          "Application Portal Authentication"  
AuthType          Basic  
AuthUserFile      /etc/httpd/conf/httpd.passwd
```

```
require          valid-user
</Location>
```

This will delegate any requests to `https://xxxxx/lap/` to the WebWare application server located on the same machine as the Web server using the tcp port 8086. Port can be configured in the `[instance dir]/Configs/AppServer.config` file.

When the Apache webserver is restarted, the WebWare default page should be shown when directed to `https://xxxxx/lap/`.

Authentication is done using a technique called fake basic authentication. Basically, authentication by X509 certificates is done in a similar way as way normal basic authentication in the Web server. The difference is in the way the authentication file is constructed. Instead of username and password pairs, the DN:s of the users are listed in the file and a special hash, `xxj31ZMTZzkVA`, is used to indicate fake-basic authentication. A typical password file (`/etc/httpd/conf/httpd.passwd`) is shown below:

```
/O=Grid/O=NorduGrid/OU=lunarc.lu.se/CN=Jonas Lindemann:xxj31ZMTZzkVA
...
```

## 9 Installing the Lunarc Application Portal

The Lunarc Application Portal is intended to be installed in a WebWare application instance. The basic procedure is to download and extract the source packages to the root of the instance directory.

```
[root@grid lap]# cd [webware-instance]
[root@grid lap]# tar xvzf lapportal-0.9.0-knowarc1.tar.gz
```

## 10 Configuring the portal

Configuration settings for the portal can be found in the `[instance dir]/LapConfig/LapSite.py` and should be modified to match the local setup. A typical configuration is shown below:

```
Application = {
    "ContextName": "context"
}

Logging = {
    "LogFile": "/var/log/lap/lap.log",
    "WebWareLogFile": "/var/log/lap/lap_ww.log",
    "LogLevel": "DEBUG"
}

Dirs = {
    "PluginDir": "/opt/lap/context/Plugins",
    "SessionDir": "/var/spool/lap",
    "WebWareDir": "/opt/Webware-1.0.2",
    "AppWorkDir": "/opt/lap",
    "DependsDir": "/opt/lap/depends",
    "NordugridDir": "/opt/nordugrid"
```

```

}

Appearance = {
    "WelcomeMessage": "Welcome to the LUNARC application portal",
    "WebSiteName": "LUNARC",
    "LogoImage": "images/logo.png",
    "LogoImageWidth": "445px",
    "LogoImageHeight": "86px"
}

Admin = {
    "VOAdmin": "/O=Grid/O=NorduGrid/OU=byggmek.lth.se/CN=Jonas Lindemann",
    "VOSites": ["1.2.3.4"],
    "UserAdmin": "/O=Grid/O=NorduGrid/OU=byggmek.lth.se/CN=Jonas Lindemann",
    "UserSites": ["1.2.3.4"],
    "UserListEnabled": False
}

System = {
    "SMTPServer": "mail.lth.se",
    "ServerUser": "root",
    "ServerGroup": "root",
    "ServerPIDFile": "/var/run/lap.pid",
    "ARCInterface": ""
}

```

Please make sure that the following directories are created:

- /var/spool/lap
- /var/lap/log

The next step is to modify the Launch.py script which is a Python script to bootstrap the WebWare application instance.

```

#!/usr/bin/python

# You can pass several parameters on the command line
# (more info by running this with option --help)
# or you can modify the default values here
# (more info in WebKit.Launch):

workDir = None
webwareDir = '/opt/Webware-1.0.2'
libraryDirs = ['/opt/lap/LapConfig', '/opt/lap/webkit_ext', '/opt/lap/webkit', '/opt/lap/context', '']
runProfile = 0
logFile = None
pidFile = None
user = None
group = None

import sys
sys.path.insert(0, webwareDir)
sys.path.insert(0, '/opt/lap/webkit_ext')
...

```

The final step of the configuration is to modify and install the init-script (for RedHat based systems). The init-script is located in `[instance_dir]/webkit`. There are 3 variables that must be added in this file. The variables are shown below:

```
...
# Unset the following variable if you want to store the
# pid and log files as the user running the start script
# (usually root) or set it if you want these files to be
# written after switching to WEBWARE_USER:WEBWARE_GROUP.
LAUNCH_AS_WEBWARE="yes"

# Additional options -u or -0 to be passed on to Python:
PYTHONOPTS=
# Additional libraries to be included in the Python path:

export GLOBUS_LOCATION=/opt/globus
export LD_LIBRARY_PATH=/opt/arc-trunk/lib64:/opt/arc-trunk/lib:/opt/arc-trunk/lib/arc:/opt/arc-trunk
export PYTHONPATH=/opt/arc-trunk/lib64/python2.4/site-packages

### END LOCAL CONFIGURATION
...
```

`/opt/arc-trunk` is the install location of ARC1.

## 11 Creating a user

The 0.9.0 version of the Lunarc Application Portal uses a user name password scheme for user authentication in addition to the X509 authentication. A special script, `createUser.py`, in the instance directory can be used to create users and directories. A user is then created as follows:

```
[root@lap trunk]# ./createUser.py
Please enter username: jonas
Please enter user password : [enter a password]
```

This procedure creates a directory in the user session directory of the portal (default `/var/spool/lap`)