

# Table of Contents

<b>UNICORE Registry Reference Card.....</b>	<b>1</b>
Functional description.....	1
Daemons running.....	1
Init scripts and options (start stop restart ... ).....	1
Configuration files location with example or template.....	1
Logfile locations (and management) and other useful audit information.....	1
Open ports.....	1
Possible unit test of the service.....	1
Where is service state held (and can it be rebuilt).....	1
Cron jobs.....	2
Security information.....	2
Access control Mechanism description (authentication & authorization).....	2
How to block/ban a user.....	2
Network Usage.....	2
Firewall configuration.....	2
Security recommendations.....	2
Security incompatibilities.....	2
List of externals (packages are NOT maintained by Red Hat).....	2
Other security relevant comments.....	2
Utility scripts.....	3

# UNICORE Registry Reference Card

## Functional description

The UNICORE Registry is a specially configured instance of a UNICORE/X server, providing a special service called a "Registry". This is used by clients to find required services. It is populated by the UNICORE/X servers, and the content is held up-to-date using entry lifetime mechanisms.

## Daemons running

The UNICORE Registry is a single process.

## Init scripts and options (start|stop|restart|...)

The service can be started with `/etc/init.d/unicore-registry {start|stop|restart}`.

## Configuration files location with example or template

The configuration files are located in `/etc/unicore/registry`

- `wsrflite.xml` : keystore/truststore locations and passwords, gateway location, host/port, deployed web services, service persistence configuration
- `uas.properties` : some service container configuration (startup code, etc). AuthZ attribute source configuration
- `xacml2Policies/*.xml` : XACML security policies
- `logging.properties` : log4j logging configuration

## Logfile locations (and management) and other useful audit information

The log files will be written to `/var/log/unicore/registry/registry.log` Logfiles are by default rolled over daily. Details can be controlled in the `logging.properties` file

## Open ports

- the web server port, configured in the `wsrflite.xml` file (default: 7778).

## Possible unit test of the service

Unit tests are part of the build procedure and executed automatically.

## Where is service state held (and can it be rebuilt)

Service state is held in a configurable database. By default, the data is kept on the file system (using an embedded database engine). The service state is held in `/var/lib/unicore/registry`

## Cron jobs

None

## Security information

### Access control Mechanism description (authentication & authorization)

Users are authenticated by the UNICORE gateway. Authorization is performed by UNICORE/X in the following way

- based in the user's identity, authz attributes are fetched from the configured sources
- based on these attributes, an XACML callout is made to check that the current operation (web service call) is allowed
- if not allowed, an "Access denied" fault is thrown

### How to block/ban a user

Revoke the certificate. It is possible to ban a user by using the XACML policy check. Ie, add a rule denying access to role "banned" and assign that role to the user that should be banned.

## Network Usage

A UNICORE Registry will connect to

- UNICORE gateway(s)
- AuthZ attribute services (UVOS, XUADB, SAML-VOMS) depending on configuration

## Firewall configuration

- see above for outbound connections

## Security recommendations

Do not run as root.

## Security incompatibilities

None known.

## List of externals (packages are NOT maintained by Red Hat)

n/a

## Other security relevant comments

n/a

## Utility scripts

n/a

-- BerndSchuller - 18-Mar-2011

---

This topic: EMI > UNICOREregistrySRC

Topic revision: r4 - 2011-07-21 - BerndThomasSchullerExCern



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)