UNICORE REGISTRY MANUAL

UNICORE Team

Document Version:	1.0.0
Component Version:	8.3.0
Date:	15 12 2021

Contents

The UNICORE Registry server provides information about available services to clients and other services. It is a specially configured UNICORE/X server, so please make sure to refer to the general UNICORE/X manual as well.

Multiple UNICORE/X sites can share a Registry, greatly simplifying the use of UNICORE services. Since such a registry is vital to the functioning of a UNICORE-based federation, you can have more than one.

For more information about UNICORE visit https://www.unicore.eu.

1 Installation

1.1 Prerequisites

To run the Registry, you need the OpenJDK or Oracle Java (JRE or SDK), in version 8 or later.

UNICORE has been most extensively tested on Linux-like systems, but runs on MacOS/X as well.

Please note that

- to integrate into secure production environments, you will need access to a certificate authority and generate server certificates for all your UNICORE servers.
- to make your UNICORE servers accessible outside of your firewalls, you should setup and configure a UNICORE Gateway.

1.2 A note on paths

The Registry can be installed either from a Linux package (i.e. RPM or deb), from a tar.gz or even from the UNICORE core server bundle package.

Note
Using the Linux packages, you can install only a single Registry instance per machine (without manual changes).

The following table gives an overview of the file locations for both tar.gz and Linux packages.

Table 1: Directory Layout

Name in this manual	tar.gz, zip	rpm	Description
CONF	<basedir>/conf/	/etc/unicore/registry	Config files
LIB	<basedir>/lib/	/usr/share/unicore/registry/lib	Java libraries
LOG	<basedir>/log/	/var/log/unicore/registry/	Log files
BIN	<basedir>/bin/	/usr/sbin/	Start/stop scripts

2 Registry configuration

A Registry is running in a "normal" UNICORE/X container, however, you should use a dedicated UNICORE/X instance for the Registry, making sure no other services are running.

Thus, most of the UNICORE/X documentation regarding access control, keystores, etc also applies to the Registry. Please, make sure to read the UNICORE/X documentation as well.

2.1 Registry configuration (CONF/uas.config)

Apart from hostname, port, and other properties, the uas.config file must contain the following entry

```
container.feature.Registry.mode=shared
```

This setting configures the container to operate as a shared Registry.

2.2 Starting and stopping

The Registry is started and stopped like any other UNICORE/X container using the scripts in the "bin" folder.

2.3 Access control

It is absolutely VITAL that the Registry only contains trusted entries. Therefore access control is enabled by default. It is configured in CONF/uas.config

```
container.security.accesscontrol.Registry=true
```

("true" by default)

This will check the security policy (CONF/xacml2Policies/*.xml) for each request. By default, this policy allows to add entries only for callers with the role "server".

If using an XUUDB or other attribute source, you will need to add the certificates / DNs of all servers wishing to publish into the registry as having the role "server". Please check the UNICORE/X documentation on how to do that.

2.4 User / server authentication

While users can read registry content without needing to be authenticated, servers MUST be authenticated and mapped to role "server" to be able to write to the Registry.

Servers using the XML/SOAP interface are authenticated via their X509 certificate.

To accept servers, the REST interface must be configured for X509 authentication as well.

As an example the following configuration will achieve this

```
#
# Authentication for the REST interface
#
container.security.rest.authentication.order=X509
container.security.rest.authentication.X509.class=eu.unicore. ↩
    services.rest.security.X509Authenticator
```

For further details we refer also to the UNICORE/X documentation on authentication and REST services.

2.5 Gateway configuration

If running the Registry behind a Gateway, you'll need to add an entry to the Gateway's site list file (connections.properties) that points to your Registry server. Another option is to use dynamic registration. In the following, we assume the Registry is named "REGISTRY".

2.6 UNICORE/X configuration

To publish the services in a shared registry, configure the address of the registry in uas.config :

```
# switch on use of external registry
container.externalregistry.use=true

# URL
container.externalregistry.url=https://...

# optionally you can have more registries
container.externalregistry.url.2=https://...
```

The entries in the global Registry are updated at a specified interval. To control this interval, edit a property in CONF/container.properties

```
# default termination time for registry entries in seconds
container.wsrf.sg.defaulttermtime=1800
```

2.7 Client configuration

Clients will require the URL of a Registry. For example, in the UCC preferences file (supply the correct values for your setup):

```
registry=https://gwhost:port/REGISTRY/rest/registries/ ↩
    default_registry
```