



# UNICORE COMMON INFORMATION SERVICE (CIS) MANUAL

UNICORE Team

---

Document Version:	1.3.0
Component Version:	1.3.0
Date:	30 08 2013

---

This work is co-funded by the EC EMI project under the FP7 Collaborative Projects Grant Agreement Nr. INFSO-RI-261611.



## Contents

<b>1 Overview</b>	<b>1</b>
<b>2 Features</b>	<b>1</b>
2.1 Server . . . . .	1
2.2 Client . . . . .	1
<b>3 Getting Started</b>	<b>1</b>
3.1 Prerequisites . . . . .	2
3.2 Installation . . . . .	2
<b>4 Configuration</b>	<b>2</b>
4.1 Services (CONF/wsrfite.xml) . . . . .	2
4.2 General (CONF/cis.config) . . . . .	3
4.3 Security . . . . .	5
4.4 UNICORE Command Line Client (UCC) Plug-in . . . . .	5

## 1 Overview

The UNICORE CIS is an information service which captures information about the location, storage, compute, as well as the running jobs on a unicon site. The site information is managed in the OGF's GLUE2 format, which can be viewed either on a Web browser or by executing the UCC commands. For the detailed information about the configuration and deployment.

## 2 Features

### 2.1 Server

- Web services interface to execute CIS functions
- Authentication and authorisation mechanisms based on UNICORE/X
- Native XML Database with Lucene support for storing and querying the information

### 2.2 Client

#### 2.2.1 Web

- Showing site location as a marker on Google Maps
- Site listing
- Full text search based on keywords

#### 2.2.2 UCC

- Viewing collection of UNICORE site information
- Managing common information providers (CIP)
- Execute XPath and XQuery on stored site information
- Discover the CIS "instance" from the Global UNICORE Registry

For more information about UNICORE visit <http://www.unicore.eu>.

## 3 Getting Started

The Common Information Service (CIS) is based on UNICORE/X, eXistDB XML database and optionally Spring framework. It consists of a central service, an information provider component (CIP) and pluggable extensions to the UNICORE Command Line Client (UCC). This guide describes the installation and configuration of the CIS server.

### 3.1 Prerequisites

In order to install CIS, the Oracle or OpenJDK Java 6 (JRE or SDK) is required.

- Any Linux based operating system
- See [prerequisites](#) for further prerequisites

### 3.2 Installation

CIS is bundled in a tar.gz file, in order to install the bundle, unpack the archive in a directory of your choice. The hostname, paths, and ports can be set in the configuration files, which resides in the **CONF** directory. Here **CONF** refers to the CIS configuration directory regardless of the distribution type. In case of binary distribution it can be found in the main CIS folder, and in case of RPM distribution the **/etc/unicore/cis**.

Since CIS has been implemented using UNICORE, therefore most of the [documentation](#) regarding access control, keystores, service configuration akin to the UNICORE/X manual.

The [configuration options](#) intends to give an overview of the configuration.

## 4 Configuration

The [table](#) provides description about the configuration files. The *cis.config* is being used instead of the *uas.config*, however the relevant files in the CIS's domain are:

- wsrf-lite.xml
- logging.properties
- xacml2Policies/\*.xml
- xacml2.config
- simpleuudb

### 4.1 Services (CONF/wsrf-lite.xml)

Besides hostname, port, and security properties, the wsrf-lite.xml file (if undefined) should contain the following CIS specific declarations.

```
<!-- declaration of the CIS service -->
<service name="CISRegistry" wsrf="false">
  <interface class="de.fzj.unicore.cis. ←
    RegistryService" />
```

```

        <implementation class="de.fzj.unicore.cis.impl. ↵
            registry.CISRegistryServiceImpl" />
    </service>

    <!-- the local registry services' declarations -->
    <service name="ServiceGroupEntry" wsrf="true" persistent=" ↵
        true">
        <interface class="de.fzj.unicore.wsrf-lite.xmlbeans. ↵
            sg.ServiceGroupEntry"/>
        <implementation class="de.fzj.unicore.wsrf-lite. ↵
            xmlbeans.registry.LocalRegistryEntryHomeImpl"/>
    </service>
    <service name="Registry" wsrf="true" persistent="true">
        <interface class="de.fzj.unicore.wsrf-lite.xmlbeans. ↵
            sg.Registry"/>
        <implementation class="de.fzj.unicore.wsrf-lite. ↵
            xmlbeans.registry.LocalRegistryEntryHomeImpl"/>
    </service>

```

Please refer to the wsrf-lite container the [documentation](#) to get an overview of all the configuration options.

## 4.2 General (CONF/cis.config)

Following table describes the CIS specific properties defined in the **CONF/cis.config**.

Property name	Type	Default value / mandatory	Description
<code>cis.cip.externalRegistry.address</code>	string	<code>https://example.org:8080/REGISTRY/services/Registry?res=default_registry</code>	The address of the external uncore registry. Specifying the address would automatically fetch the CIP URLs from a UNICORE's external registry
<code>cis.cip.externalRegistry.autoRegister</code>	[true, false]	false	The boolean flag, if enabled (set to <i>true</i> ) will contact remote uncore registry to fetch all the CIP endpoint addresses

Property name	Type	Default value / mandatory	Description
<code>cis.cip.listPath</code>	string	-	Location of an xml file, should contain a listing of all the CIP URLs. It is an alternative to the external registry source
<code>cis.cip.pollInterval</code>	integer number	10	Defines an interval in seconds to fetch uncore information from given CIP URL
<code>cis.dmon.enable</code>	[true, false]	false	D-MON project specific information, setting to <i>true</i> will allow the CIS server to fetch information regarding the computing shares from the given D-MON central database
<code>cis.existdb.embedded.confPath</code>	string	CONF/xmlldb/conf.xml	Location of the eXist-DB (embedded xml database) specific configuration file
<code>cis.existdb.embedded.indexPath</code>	string	CONF/xmlldb/collectio- n.xconf	Location of the embedded eXist-DB (native xml database), provide configuration options for the lucene index over xml database
<code>cis.existdb.onStart.removeall</code>	[true, false]	false	To control whether the contents of the database/index should be removed on CIS server start-up
<code>cis.existdb.remote.address</code>	string	xmlldb:exist://example.org:8090/exist/xmlrpc/db	URL of a remote eXist-DB (native xml database) server
<code>cis.springbeans.path</code>	string	CONF/cis-beans.xml	Location of the Spring's application context file containing the JDBC beans description. The file contains D-MON's database properties

Property name	Type	Default value / mandatory	Description
<code>cis.webAddress</code>	string	<code>http://-0.0.0.0-:8989</code>	The address to view the GUI on a Web browser, the URL format should be <i>http://example.org:8989</i>
<code>cis.webapp.path</code>	string	<code>CONF/we-bapp</code>	Path to the <i>webapp</i> directory, contains all the Web GUI content shown on the browser

**CIP Publishing Priority** As there are two alternate sources to get the CIP URLs, if both are enabled, the *External Registry* will take the priority over the *File* based mechanism.

**Web GUI Settings** The anonymous Web GUI port defined above should be different from the one specified in the `CONF/wsrf-lite.xml`.



#### Caution

It is recommended to leave the database and Web application configurations intact unless new Java servlets and/or XML database are required.

### 4.3 Security

The authorisation and authentication carried out in CIS based entirely on UNICORE's security mechanisms. Therefore similar configuration options (as defined in `uas.config`) should be used to enable the authentication and authorisation.

For detailed guidelines, please refer to the security related sections of the UNICORE/X documentation:

- [Security Concepts](#)
- [Authorisation](#)
- [PDP](#)
- [Security Policies](#)

### 4.4 UNICORE Command Line Client (UCC) Plug-in

As mentioned earlier, the UCC can be used to view and search the site information as well as manage the CIPs remotely.

#### 4.4.1 Installation

UCC must be installed (e.g. in `/UCC-INSTALL-DIR/`) before executing the CIS specific commands.

##### *Setting up*

- Add the jar-files from CIS' s *lib* directory to the ucc's lib directory `UCC-INSTALL-DIR-  
/lib`
- Append the contents of `CONF/extensions+` to the `/UCC-INSTALL-DIR/extensions`. This will extend the existing ucc's command list and will show the commands under **Grid Information** - the ucc's command group
- Add CIS URL property to your ucc's "preferences file", usually this resides inside **conf/preferences**

```
cisurl=http(s)://CIS-HOSTNAME:CIS-PORT/services/CISRegistry
```

#### 4.4.2 Commands

The following ucc commands are available for accessing the CIS

- Add a CIP (Information Provider/Source) to the CIS's domain; it will then collect the site information from that specific site. The `cip-url` is the corresponding logical unicorex-url or path to Glue2 XML file

```
ucc cis-addcip -c <your userprefs.txt file> -i <cip-url> -t <file| ↵  
ws>
```

- List all CIPs registered at the CIS

```
ucc cis-showallcip -c <your userprefs.txt file>
```

- List the information collected from all the registered CIPs (all UNICORE Sites)

```
ucc cis-showallinfo -c <your userprefs.txt file>
```

- Remove a CIP from the CIS

```
ucc cis-removecip -c <your userprefs.txt file> -i <cip-url>
```



- Query the CIS using XQuery/XPath format

```
ucc cis-queryinfo -c <your userprefs.txt file> -q <query expression <←  
>
```

- Returns an aggregated view of all the sites information as a single GLUE2 document

```
ucc cis-showdoc -c <your userprefs.txt file>
```

- Free text search on UNICORE site information collection

```
ucc cis-search -c <your userprefs.txt file> -f <keyword(s)>
```

- Listing all the registered CIS in UNICORE Global Registry (if the **cis-url** is unknown)

```
ucc list-cis -c <your userprefs.txt file>
```

For further information about installing and using the ucc can be found under [UCC Manual](#)