Installing OSCARS 0.6

1 Overview 3

1.1 Introduction 3

1.2 Hardware are software requirements 3

1.2.1 System Requirements 3

1.2.2 Network Requirements 3

1.2.3 Firewall Requirements 3

1.2.4 Third-Party Library and Package Requirements 3

2 Preparing your environment 3

3 Installing the OSCARS Inter-domain controller (IDC) Software 4

3.1 Installing OSCARS .6 for the first time 4

3.2 Upgrading an Existing IDC from Version 0.5.3.X to 0.6 4

3.2.1 Prepare your environment 4

3.2.2 Install OSCARS v6 and upgrade 0.5 database 5

3.2.3 Migrating certificates 5

3.2.4 Firewall requirements 6

3.2.5 Update Configuration Files 7

4 Starting/Stopping OSCARS 17

5 Configuration and customization 18

5.1 Create the first WBUI user 18

5.2 Verifying that your basic installation was successful 19

5.3 Define your local domain 19

5.4 Define your Topology 19

5.4.1 Creating an XML Topology Description 19

5.4.2 Configure Topology information 20

5.5 Changing ports used 21

5.6 Changing your DB password 22

6 Creating and Managing User Accounts 24

6.1 Adding Users to the Database 24

6.1.1 Creating New User Accounts 24

6.1.2 Modifying Users 25

6.1.3 Deleting Users 25

6.2 Managing X.509 Certificates from Users and Other IDCs 25

7 Using the ION UI 25

7.1 Starting/Stopping ION UI Service 25

7.2 Configuring the ION UI Service 26

7.2.1 PerfSonar-PS components 26

7.2.2 Database (MySql) password 27

7.2.3 Port number 27

8 Inter-domain configuration 27

8.1 Making your IDC Aware of Other Domains 27

8.2 IDC certificates for sending inter-domain requests 28

8.3 Test your sample multi-domain installation 28

9 Customizing logging for your deployment 28

# Overview

## Introduction

This document intends to guide a user in installing OSCARS 0.6. This document includes details to install, configure and customize the installation for your site.

## Hardware are software requirements

### System Requirements

* A Linux host running [KVM](http://www.linux-kvm.org/page/Main_Page)
* 8GB of memory recommended
* 30GB of hard disk space

### Network Requirements

### Firewall Requirements

OSCARS 0.6 uses a TCP ports 8443 (WBUI), 9001, 9013 and 9196(IONUI) by default.

### Third-Party Library and Package Requirements

TBD

# Preparing your environment

OSCARS can be installed as a set of RPMs now. This section lists a set of basic steps you need to be able to install OSCARS software using these RPMs.

* + Create a file which has the Internet2 repository details:

|  |
| --- |
| cd /etc/yum.repos.d  cat >> Internet2-OSCARS.repo  # Name: Internet2 RPM Repository  # URL: http://software.internet2.edu  [Internet2-OSCARSDev]  name = Internet2 RPM Repository - software.internet2.edu - OSCARS DEV  baseurl = http://software.internet2.edu/branches/andy-oscars-testing/rpms/i386/main/  enabled = 1  protect = 0  gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Internet2  gpgcheck = 1 |

* + Make sure the Internet2 key is installed.

|  |
| --- |
| rpm --import <http://software.internet2.edu/rpms/RPM-GPG-KEY-Internet2>) |

# Installing the OSCARS Inter-domain controller (IDC) Software

## Installing OSCARS .6 for the first time

1. Install Oscars packages.

|  |
| --- |
| yum install oscars |

This step ensures that all necessary OSCARS packages and requirements like java and Mysql are also installed.

Currently, the only recommended version of java is 1.6.0-openjdk. If your system is pre-installed with Java, be sure it is this version. TBD: Check/recommend system setup if this is the case, or if this needs to be mentioned here.

1. Install a PSS package of your choice. The currently available options are
   * oscars-pss-dragon
   * oscars-pss-eompls
   * oscars-pss-openflow
   * oscars-pss-stub

|  |
| --- |
| yum install <your choice of pss> |

1. Install tools and scripts that help you with configuration steps.

|  |
| --- |
| yum install oscars-tools |

At this stage, you have completed installing OSCARS, and can begin configuring your installation. Proceed to section TBD to continue.

## Upgrading an Existing IDC from Version 0.5.3.X to 0.6

The steps to upgrade an existing IDC From Version 0.5.3.X to Version 0.6 are discussed in the sections below.

### Prepare your environment

* Backup OSCARS v5
  + mv $OSCARS\_HOME /usr/local/oscars\_home\_v5
  + mysqldump -u root -p --all-databases > oscars-0.5-backup-all.sql
* Prepare your environment:
  + Find the existing mysql 'root' and 'oscars' passwords. If the MySql root password is not null/empty, change it to use no password and retain it to be so until after the RPM installation is complete.
  + Check that step [Preparing your environment](#_Preparing_your_environment) has been completed

### Install OSCARS v6 and upgrade 0.5 database

Once your environment is ready, you can start the OSCARS 0.6 installation procedure. available RPMs will also automatically upgrade your current 0.5 database. The steps involved are:

1. Install Oscars packages.

|  |
| --- |
| yum install oscars |

This step checks for the existence of 0.5 tables and then executes steps to copy your existing AAA and Reservations related data into relevant 0.6 database tables. Please note these:

* 1. RPMs do not perform any “merge” of data in case you have both 0.5 and 0.6 tables
  2. If you have an already installed 0.6 version, then your database data will remain untouched.

1. Install a PSS package of your choice. The currently available options are
   * oscars-pss-dragon
   * oscars-pss-eompls
   * oscars-pss-openflow
   * oscars-pss-stub

|  |
| --- |
| yum install <your choice of pss> |

1. Install tools and scripts that help you with configuration steps.

|  |
| --- |
| yum install oscars-tools |

#### Moving your installation to a new host?

If you wish to move your 0.5 data to a new host and install 0.6 over it, the databases of interest to you are “aaa” and “bss”. You can dump these databases and import them on to your new installation. The “aaa” databsase contains user (and his attributes) and institution/site data, while the “bss” database has reservations data and history. Once you have this done, you can run the regular RPM installation (follow steps 1 through 3 in section [above](#_Install_OSCARS_v6)). At the end of your 0.6 installation, you may drop databases “aaa”,“bss” on the host.

At this stage, your installation process is completed. The steps that follow pertain to moving your configuration data.

### Migrating certificates

These are the steps to port the Certificates used by your OSCARS 0.5 to the new 0.6 installation setup:

1. Copy your 0.5 keystore OSCARS.jks and ssl-keystore.jks. These key stores are usually in location $OSCARS\_HOME/conf/axis-tomcat.
2. Drop the existing 'mykey' from /etc/oscars/key stores/oscars.jks.

|  |
| --- |
| keytool –delete –alias mykey –keystore /etc/oscars/keystores/oscars.jks |

1. Import OSCARS.jks into Oscars.jks

|  |
| --- |
| keytool -importkeystore -srckeystore <path\_to\_OSCARS.jks> -destkeystore /etc/oscars/keystores/oscars.jks |

1. Import ssl-keystore.jks into Oscars.jks

|  |
| --- |
| keytool -importkeystore -srckeystore <path\_to\_ssl-keystore.jks> -destkeystore oscars.jks |

1. Clean up oscars.jks to make sure it only contains only one privateKey entry. This should be the key whose alias name was mentioned in your 0.5’ configuration file rampConfig.xml. For ease of reading, let us refer to this key as “newkey” for the remainder of this section.
2. Change the password for this alias from “password” to the default password “changeit” used by the 0.6 installation.

|  |
| --- |
| keytool -keypasswd -alias newkey -keystore /etc/oscars/keystores/oscars.jks |

1. Add the certificate chain of the CA that signed “newkey” into localhost.jks.

To copy certificates, use *keytool -export* and then -*import*. These commands are along the line of:

|  |
| --- |
| keytool -export -keystore OSCARS.jks -file saveCAs.out |

And then, to import:

|  |
| --- |
| keytool –import –file saveCAs.out –keystore /etc/oscars/keystores/localhost.jks |

You will be prompted to verify whether you trust this CA when you run the above command.

#### Change references to certificate alias

Change certificate alias from 'mykey' into the alias defined in rampConfig.xml for OSCARSService, NotificationBridgeService and WSNBrokerService. The new alias, using the example from Section [1.4](#_Migrating_certificates), would be “newkey”. To do so, follow these steps:

* + Open WSNBrokerService/conf/client-cxf-http.xml
  + Find lines like

<entry key="ws-security.signature.username" value="mykey" />

* + Replace “mykey” with the certificate alias from rampConfig.xml
  + Repeat this procedure in files:

|  |
| --- |
| /etc/oscars/OSCARSService/conf/client-cxf-ssl.xml  /etc/oscars/OSCARSService/conf/client-ut-cxf-ssl.xml  /etc/oscars/OSCARSService/conf/server-cxf-ssl.xml  /etc/oscars/OSCARSService/conf/client-cxf-DEV.xml  /etc/oscars/OSCARSService/conf/client-cxf-http.xml  /etc/oscars/OSCARSService/conf/server-cxf-http.xml  /etc/oscars/NotificationBridgeService/conf/client-cxf-ssl.xml  /etc/oscars/NotificationBridgeService/conf/client-cxf-http.xml  /etc/oscars/WSNBrokerService/conf/client-cxf-ssl.xml  /etc/oscars/WSNBrokerService/conf/server-cxf-ssl.xml  /etc/oscars/WSNBrokerService/conf/client-cxf-http.xml  /etc/oscars/WSNBrokerService/conf/server-cxf-http.xml |

### Firewall requirements

OSCARS 0.6 uses a few extra ports along with TCP ports 8080 and 8443 that OSCARS 0.5 used. To use 0.6, update your system configuration to allow access to ports 9001 and 9013.

These ports are used by OSCARSService and WSNBroker respectively. So, in the case you wish to modify the default port numbers used by OSCARS services, set your firewalls to use the ports used by OSCARSService and WSNBroker instead of 9001 and 9013.

In Linux, one could edit /etc/sysconfig/iptables to add lines similar to:

|  |
| --- |
| -A INPUT -m state --state NEW -m tcp -p tcp --dport 8443 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 9001 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 9013 -j ACCEPT |

### Update Configuration Files

To be able to successfully upgrade from Oscars 0.5.X to 0.6, a few configuration parameters need to be translated to the 0.6 set up. These are listed below.

#### Update publishTo address

Update the “publishTo” address to point to the public IP address or hostname in the OSCARSInternalService, OSCARSService and WSNBrokerService processes.

The list of files that need changes to the “publishTo” line are:

|  |
| --- |
| $OSCARS\_HOME/OSCARSInternalService/conf/config-internal.HTTP.yaml  $OSCARS\_HOME/OSCARSInternalService/conf/config-internal.SSL.yaml  $OSCARS\_HOME/OSCARSService/conf/config.SSL.yaml  $OSCARS\_HOME/OSCARSService/conf/config.HTTP.yaml  $OSCARS\_HOME/WSNBrokerService/conf/config.SSL.yaml  $OSCARS\_HOME/WSNBrokerService/conf/config.HTTP.yaml |

#### Replace any occurrences of word “localhost localdomain”

The RPM installation tries to fetch your hostname and refer to it in the various configuration files. However, if that fails due to any reason, you may be left with occurrences of the word “localhost.localdomain” in your configuration files. Replace these with your hostname/IP address.

The list of files that (may) need changes are:

|  |
| --- |
| $OSCARS\_HOME/OSCARSService/conf/server-cxf-ssl.xml  $OSCARS\_HOME/OSCARSService/conf/config.HTTP.yaml  $OSCARS\_HOME/OSCARSService/conf/server-cxf-http.xml  $OSCARS\_HOME/OSCARSService/conf/config.SSL.yaml  $OSCARS\_HOME/WSNBrokerService/conf/server-cxf-ssl.xml  $OSCARS\_HOME/WSNBrokerService/conf/server-cxf-http.xml  $OSCARS\_HOME/OSCARSInternalService/conf/config-internal.SSL.yaml  $OSCARS\_HOME/OSCARSInternalService/conf/config-internal.HTTP.yaml |

#### Configure local domain and MPLS choices

Configure your 0.6 installation to use the local domain you used in Oscars 0.5. To do so, follow these steps:

* + Open Utils/conf/config.yaml
  + Set the local domainId to be used
    - Alternatively, you can use a shell script to add your local domain ID:

|  |
| --- |
| cd $OSCARS\_DIST/tools/bin  ./idc-localdomainmod *context* |

* + Set MPLS option to either 0 or 1

#### Configure topology file details

The following two steps detail the changes needed to configure topology file details.

* Configure your installation to refer to the correct topology files and their location. To do so, follow these steps:
  + Open files TopoBridgeService/conf/config.HTTP.yaml and TopoBridgeService/conf/config.SSL.yaml
  + in the “domains” section of this file, include your domain name. Also include the location of the file you used to define its topology. For example (notice the “file” option).

|  |
| --- |
| *domains:*  *'testdomain-1':*  *source: 'file'*  *file: 'testdomain-1.xml'* |

* + Alternatively, you can use a shell script to configure your domain data:

cd $OSCARS\_DIST/tools/bin

./idc-domaininfoadd <CONTEXT>

* Configure your installation to refer to a PerfSonar Topology Server URL, if you’re using one. To do so, follow these steps:
  + Open files TopoBridgeService/conf/config.HTTP.yaml and TopoBridgeService/conf/config.SSL.yaml
  + Include your topology server name and location in the “domains” section of this file. For example (notice that source is “topoServer”, and the section “servers” is being used).

|  |
| --- |
| *'\*':*  *source: 'topoServer'*  *servers: [ 'ts-server.domain.org' ]* |

* + Alternatively, you could use these tools to include a Perfsonar TS URL:

cd $OSCARS\_DIST/tools/bin

./idc-toposerveradd <CONTEXT>

#### Change DB salt string

Edit $OSCARS\_HOME/AuthNService/conf/authN.\*.yaml

Look for lines like that specify the “salt” string value, like

|  |
| --- |
| salt: 'aa' |

Change this value to ‘os’, so that final string is:

|  |
| --- |
| salt: 'os' |

#### Configure PSS

TBD: This section is a placeholder containing data from wiki, and Xi’s information to upgrade ION. Needs to be finalized.

See the PSS section in “Move data from Oscars.properties” below to port values to OSCARS 0.6. These pages offer more information and instructions:

<http://code.google.com/p/oscars-idc/wiki/DragonPSS> <http://code.google.com/p/oscars-idc/wiki/EoMPLSPSS> <http://code.google.com/p/oscars-idc/wiki/EoMPLSPSSConfigFiles> <http://code.google.com/p/oscars-idc/wiki/PSSFrameworkBuildingBlocks> <http://code.google.com/p/oscars-idc/wiki/PSSCommonFiles>

##### Update EomplsPSS config files

If you are using EomplsPSS, you could follow these quick instructions for upgrading to 0.6:

1. Modify config.\*.yaml

Set these parameter values:

|  |
| --- |
| stub = false  performVerify = true |

1. Modify config-connector-directory.yaml
   * + Check for these values to be correct:

|  |
| --- |
| privkeyfile  passphrase |

* + - Move privkey file (for Juniper ssh) to where config-connector-directory.yaml points to

1. Update all config-\*.yaml files using info from your topology file.
   * + Update device-addresses from <address>
     + ifce-addresses from '##'
     + models -> use juniper-mx

#### Move data from Oscars.properties

The oscars.properties file is the main location that the OSCARS 0.5.X IDC refers to for many installation-specific settings. These include properties used to access the MySQL database, AAA, the perfSONAR Lookup Service etc. Oscars 0.6 does not have an oscars.properties file, and the various properties are distributed in relevant, service related files.

This section discusses how these values could be ported from OSCARS 0.5 into the Oscars 0.6 configuration files. The section is divided into subsections detailing the properties from the 0.5.X version, and maps them to their 0.6 counterparts.

##### General parameters

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter name in Oscars 0.5*** | ***Example value*** | ***0.6 Destination file*** | ***Destination field*** |
| idc.url | http://your- idc:8443/axis2/services/OSCARS | $OSCARS\_HOME/OSCARSService/conf/config.HTTP.yaml  $OSCARS\_HOME/OSCARSService/conf/config.SSL.yaml | public:  publishTo |

##### MySQL Database Properties

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter name in Oscars 0.5*** | ***Example value*** | ***0.6 Destination file*** | ***Destination field*** |
| hibernate.connection.username | Oscars | * $OSCARS\_HOME/AuthNService/conf/authN.SSL.yaml: * $OSCARS\_HOME/AuthNService/conf/authN.HTTP.yaml * $OSCARS\_HOME/AuthZService/conf/authZ.HTTP.yaml: * $OSCARS\_HOME/AuthZService/conf/authZ.SSL.yaml: * $OSCARS\_HOME/ResourceManagerService/conf/config.SSL.yaml: * $OSCARS\_HOME/ResourceManagerService/conf/config.HTTP.yaml * $OSCARS\_HOME/IONUIService/conf/config.SSL.yaml * $OSCARS\_HOME/IONUIService/conf/config.HTTP.yaml | username |
| hibernate.connection.password | mypass | “ | password |
| hibernate.monitor | 1 | No equivalent | No equivalent |

##### Authentication, Authorization, and Accounting (AAA) Properties

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name in Oscars 0.5*** | ***Example value*** | ***0.6 Destination file*** | ***0.6 Destination parameter/field name*** |
| Aaa.salt | os | authNService/conf/authN.HTTP.yaml, authNService/conf/authN.SSL.yaml | salt |
| aaa.userName | oscars | WBUIService/conf/config.XXX.yaml | Look for section “wbui”. Then find field “userName” |
| aaa.sessionName | oscarssess |  | Look for section “wbui”. Then find field “sessionName” |
| aaa.secureCookie | Use SSL by default | WBUIService/conf/config.HTTP.yaml  WBUIService/conf/config.SSL.yaml | secureCookie  You may choose to write into only one of the files based on whether SSL=1 (config.SSL.html) or SSL=0 (config.HTTP.html) |
| aaa.useSignalTokens | 1 | No equivalent | No equivalent |
| aaa.guestLogin | Guest | No equivalent | No equivalent |

##### Topology Exchange and Pathfinding Properties

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name in Oscars 0.5*** | ***Example value*** | ***0.6 Destination file*** | ***0.6 Destination parameter/field name*** |
| pathfinder.findPath | 1/0 (If set to 0, user must provide path) | No equivalent | No equivalent |
| pathfinder.pathMethod | Perfsonar, terce, static | No equivalent | No equivalent |
| Pathfinder.pathMethod.loc | terce,perfsonar,static | No equivalent | No equivalent |
| pathfinder.pathMethod.interdomain | Perfsonar, static | No equivalent | No equivalent |
| pathfinder.staticxml.file | /home/Oscars/static-routes.xml | No equivalent | No equivalent |
| perfsonar.domainOpacity | complete | No equivalent | No equivalent |
| perfsonar.domainOpacity. path | Complete | No equivalent | No equivalent |
| tedb.tedbMethod | Terce (for Dragon installations) | No equivalent | No equivalent |
| terce.url | http://127.0.0.1:8080/ axis2/services/TERCE | No equivalent | “ |
| topo.defaultSwcapType | 1/0 (If set to 0, user must provide path) | No equivalent | “ |
| topo.defaultEncodingType | Perfsonar, terce, static | No equivalent | “ |

##### Path Setup Properties

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name in Oscars 0.5*** | ***Example value*** | ***0.6 Destination file*** | ***0.6 Destination parameter/field name*** |
| pss.method | dragon | Utils/conf/config.yaml | PSSChoice |
| pss.dragon.password | Dragon  Question: Recommend setting in deployment directory or PSS source? | DragonPSS/config/config-connector-directory.yaml | Look for “id: “dragon-vlsr”, and then “cliPassword” under “params” |
| pss.dragon.ssh.portForward | 1 | “ | Look for “id: “dragon-vlsr”, and then “sshPortForward” under “params” . Translate 1=true and 0=false |
| pss.dragon.setERO | 0 | “ | sshER0 |
| pss.dragon.tunnelMode | 0 | “ | tunnelMode |
| pss.dragon.ssh.user | oscars | “ | sshUser |
| pss.dragon.ssh.key | /home/oscars/.ssh/id\_rsa | “ | sshKey |
| pss.dragon.remotePort | 2611 | “ | remotePort |
| pss.dragon.nodeId3 | 127.0.0.1. | “ | localAddress |
| pss.dragon.nodeId.ssh | 192.168.2.4 | DragonPSS/config/config-device-addresses.yaml | nodeId.vlsr  \*note: nodeId is mapped from topology xml node <address> |
| pss.dragon.nodeId.ssh.port | 22 | “ | sshPort |
| pss.dragon.promptPattern | vlsr | “ | promptPattern |
| pss.dragon.hasNarb | 1 | “ | hasNarb. Translate 1=true and 0=false |
| pss.dragon.delay | 30 | “ | No equivalent |
| pss.dragon.inservice\_checks | 24 | “ | No equivalent |
| pss.dragon.inservice\_check \_delay | 5 | “ | No equivalent |

##### Notifications

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name in Oscars 0.5*** | ***Example value*** | ***0.6 Destination file*** | ***0.6 Destination parameter/field name*** |
| notify.observer.N | net.es.oscars.notify.EmailObserver | NotificationBridgeService/conf/config.HTTP.yaml  NotificationBridgeService/conf/config.SSL.yaml | Look for “Observers:” , and then “-class:”.  Then, choose relevant one from  net.es.oscars.notificationBridge.observers.EmailObserver  or WSNObserver |
| notify.ws.broker.url | https://your- idc.net:8443/axis2/services/OSCARSNotify | WSNBrokerService/conf/config.HTTP.yaml  WSNBrokerService/conf/config.SSL.yaml | public:  publishTo |
| notify.ws.broker.url.private | https://10.0.0.2:8443/axis2/services/ OSCARSNotify | WSNBrokerService/conf/config.HTTP.yaml  WSNBrokerService/conf/config.SSL.yaml | publishTo |
| notify.ws.broker.registerRetryAttempts | 10 | No equivalent. | No equivalent. |
| notify.ws.broker.seco ndsBetweenRegistrationRetries | 60 | No equivalent. | No equivalent. |
| notifybroker.subscrip tions.maxExpireTime | 3600 | WSNBrokerService/conf/config.HTTP.yaml  WSNBrokerService/conf/config.SSL.yaml | expiration |
| notifybroker.publishers.maxExpireTime | 3600 | No equivalent. | No equivalent. |
| notifybroker.pep.N | net.es.oscars.notify.ws.policy.IDCEventPEP | WSNBrokerService/conf/config.HTTP.yaml  WSNBrokerService/conf/config.SSL.yaml | Looks for “peps:” and then  “class:” and include 'net.es.oscars.wsnbroker.policy.IDCEventPEP' |
| mail.webmaster | webmaster@blue.pod.lan | NotificationBridgeService/conf/config.HTTP.yaml  NotificationBridgeService/conf/config.SSL.yaml | mail.from |
| Mail.recipients | user1@my.net:user2@my.net | NotificationBridgeService/conf/config.HTTP.yaml  NotificationBridgeService/conf/config.SSL.yaml | mail.to  Format as a YAML list (indented new lines starting with ‘ - ‘ |

##### External services

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name in Oscars 0.5*** | ***Example value*** | ***0.6 Destination file*** | ***0.6 Destination parameter/field name*** |
| external.service.N | Subscribe, lsRegister, lsDomainUpdate, topology | No equivalent | No equivalent |
| perfsonar.topology\_url | http://packrat.internet2.edu:8012/per fSONAR\_PS/services/topology | TopoBridgeService/conf/config.HTTP.yaml, TopoBridgeService/conf/config.SSL.yaml | “registerUrl” .Uncomment field and transfer value to it. |
| lookup.hints | <http://www.perfsonar.net/gls.root> | LookupService/conf/config.HTTP.yaml, LookupService /conf/config.SSL.yaml | perfsonar:  globalHintsFile |
| lookup.global.N | http://ndb1.internet2.edu:9990/perfS ONAR\_PS/services/gLS | LookupService/conf/config.HTTP.yaml, LookupService /conf/config.SSL.yaml | perfsonar:  globalLookupServices  Format as a YAML list (indented new lines starting with ‘ - ‘ |
| lookup.home.N | http://ndb1.internet2.edu:8005/perfS ONAR\_PS/services/hLS | LookupService/conf/config.HTTP.yaml, LookupService /conf/config.SSL.yaml | Perfsonar:  homeLookupServices  Format as a YAML list (indented new lines starting with ‘ - ‘ |
| Lookup.topology.N | http://ndb1.internet2.edu:8005/perfS ONAR\_PS/services/hLS | TopoBridgeService/conf/config.HTTP.yaml, TopoBridgeService/conf/config.SSL.yaml | Using the ‘server’ option for a domain definition in your topology configuration has the same effect. |
| Lookup.useGlobal | 1 | No equivalent | No equivalent |
| Lookup.hints.all | 1 | No equivalent | No equivalent |
| lookup.reg.location[c ountry|zipcode|state| institution|city| streetAddress|floor| room|cage|rack|shelf| latitude|longitude| continent] | Depends on field | No equivalent | Under field “public”, parameters have same name as portion under location (e.g. longitude, latitude, country, state, city, etc) |
| lookup.reg.idc.name | My network’s IDC (default=IDC URL) | OSCARSService/conf/config.SSL.yaml  OSCARSService/conf/config.SSL.yaml | Under field “public”, look for parameter “name” |
| Lookup.reg.idc.description | IDC for testing | OSCARSService/conf/config.SSL.yaml  OSCARSService/conf/config.SSL.yaml | Under field “public”, look for parameter “description” |
| lookup.reg.nb.name | My network’s NB | No equivalent | No equivalent |
| lookup.reg.nb.description | NB for testing | No equivalent | No equivalent |
| external.service.subscribe.termTimeWindow | .2 | No equivalent | No equivalent |
| external.service.subsc ribe.retryInterval | 1800 | No equivalent | No equivalent |
| external.service.subsc ribe.topics | Idc:IDC | No equivalent | No equivalent |
| external.service.[topology| lsRegister].renewTime | 1800 | No equivalent | No equivalent |
| external.service.topology.updateLocal | 1 | No equivalent | No equivalent |
| External.service.init.WaitTime | 60 | No equivalent | No equivalent |
| External.service.isDomainUpdate.refreshTime | 4200 | No equivalent | No equivalent |

##### Miscellaneous properties

|  |  |  |  |
| --- | --- | --- | --- |
| ***Parameter Name in Oscars 0.5*** | ***Example value*** | ***0.6 Destination file*** | ***0.6 Destination parameter/field name*** |
| logging.rsvlogdir | /usr/local/tomcat/logs | No equivalent | No equivalent |
| Timeout.default | 600 | No equivalent | No equivalent |
| timeout.[create| modify| cancel]Resv.confirm | 600 | No equivalent | No equivalent |
| timeout.[create| modify| cancel]Resv.complete | 600 | No equivalent | No equivalent |
| timeout.[create| teardown]Path.confir m | 600 | No equivalent | No equivalent |
| timeout.[create| teardown]Path.retry Attempts | 10 | No equivalent | No equivalent |
| timeout.teardownPat h.waitForLocalSetup Attempts | 6 | No equivalent | No equivalent |
| timeout.teardownPat h.waitForLocalSetup AttemptTime | 10 | No equivalent | No equivalent |
| policy.useService | 1 | No equivalent | No equivalent |
| policy.service.url | https://127.0.0.1/policyEngine | No equivalent | No equivalent. |
| policy.vlanFilter.scope | node | VlanPCE/conf/config.SSL.yaml  OSCARSService/conf/config.SSL.yaml | vlanScope |
| rmi.[aaa|bss| notifybroker].serverPort | 1098 | No equivalent | No equivalent |
| rmi.[aaa|bss| notifybroker].registr yHost | 127.0.0.1 | No equivalent | No equivalent |
| rmi.[aaa|bss|notifybroker].registr yPort | 1099 | No equivalent | No equivalent |
| rmi.[aaa|bss|notifybroker].registeredServerName | AAARmiServer | No equivalent | No equivalent |
| wbui.defaultLayer | 3 | No equivalent | No equivalent |

#### Modify WBUI access settings

By default, you can use your browser to login in to the WBUI at https:/yourhost.yourdomain:8443/OSCARS. You may then login using the user credentials from your 0.5 installation. Currently you can use the browser interface to manager users, create, query or list reservations.

The default configuration for the WBUI is to allow access to from all hosts. To allow web access to only the localhost, edit the files $OSCARS\_HOME/WBUIService/conf/ jetty.XXX.xml. Look for a line like:

|  |
| --- |
| <Set name="host">localhost.localdomain</Set> |

Change the <host.domain> entry, or uncomment out this line.

#### Configure Inter-domain peering

Data about peering domains used to be added into the MySQL database in 0.5.X. To continue peering with those domains, transfer this data into OSCARS 0.6. For ease of reading, let us name your domain as “your\_domain” and refer to the peer domain and host using valuesof variables peer\_domain and peer\_host.

##### Peering with an 0.6 domain

Add peering domains to the lookup service by following the steps below:

|  |
| --- |
| cd $OSCARS\_DIST/lookup  ./bin/oscars-idcadd -d $peer\_domain -p http://oscars.es.net/OSCARS/06 -l http://$peer\_host:9001/OSCARS |

Remember to run this step on both the 0.6 domains that you want to peer.

##### Peering with 0.5 domain

To peer with a 0.5 domain, run these set of steps:

* Make your current 0.6 installation aware of the 0.5 domain

|  |
| --- |
| cd $OSCARS\_DIST/lookup  ./bin/oscars-lookupadmin --cache-add -t IDC -p http://oscars.es.net/OSCARS=https://$peer\_host:8443/axis2/services/OSCARS -r controls=urn:ogf:network:domain=$peer\_domain,publisher=https://$peer\_host:8443/axis2/services/OSCARSNotify |

* Make the peer 0.5 domain aware of your 0.6 installation

On your peering 0.5 IDC, execute the following set of commands:

|  |
| --- |
| cd $OSCARS\_HOME/tools/utils/  ./idc-domainmod  Select the entry that corresponds to your\_domain. Enter https://<yourhost.yourdomain>:9001/OSCARS when prompted for URL    ./idc-servicemod  Select the entry that corresponds to your\_domain and “NB”. Enter  https:// <yourhost.yourdomain>:9013/OSCARS/wsnbroker when prompted for URL |

##### Disable hostName check

If you would like to disable the hostname checks in peer SSL certificate verification, change configuration files to do so.

Look for lines like “disableCNCheck=false” and change this to “disableCNCheck=true”.

The files that need these changes are:

|  |
| --- |
| $OSCARS\_HOME/TopoBridgeService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/DijkstraPCE/conf/client-cxf-ssl.xml  $OSCARS\_HOME/VlanPCE/conf/client-cxf-ssl.xml  $OSCARS\_HOME/PCEService/conf/pce-runtime-client-cxf-ssl.xml  $OSCARS\_HOME/PCEService/conf/nullagg-client-cxf-ssl.xml  $OSCARS\_HOME/AuthNService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/LookupService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/ConnectivityPCE/conf/client-cxf-ssl.xml  $OSCARS\_HOME/AuthZPolicyService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/WBUIService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/AuthNPolicyService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/L3MplsPCE/conf/client-cxf-ssl.xml  $OSCARS\_HOME/CoordService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/WSNBrokerService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/AuthZService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/BandwidthPCE/conf/client-cxf-ssl.xml  $OSCARS\_HOME/ResourceManagerService/conf/client-cxf-ssl.xml  $OSCARS\_HOME/PSSService/conf/client-cxf-ssl.xml |

In linux, you may run a command like

|  |
| --- |
| grep -r "disableCNCheck=\"false\"" /etc/oscars/ |cut -d: -f1 |xargs sed -i "s/disableCNCheck=\"false\"/disableCNCheck=\"true\"/g" |

Please exercise caution when you do so and check each file to see if the contents remain as you desire.

# Starting/Stopping OSCARS

OSCARS can be started or stopped using this command:

|  |
| --- |
| /etc/init.d/oscars *option*  where *option is one of “start”, “stop”, “restart”, “status” or “help”* |

Here are some important directories.

Log directory: /var/log/oscars

This location contains logs, all named according to the “service” they belong to.

$OSCARS\_HOME: /etc/oscars

This location contains configuration files.

$OSCARS\_DIST: /opt/oscars

Contains installed jars/scripts:/opt/oscars

The above command can be used to start/stop all services.

To start some services alone individually, you can use the service startup scripts in /etc/init.d.

Some of them are:

|  |
| --- |
| * /etc/init.d/oscars-api *option* * /etc/init.d/oscars-notificationBridge *option* * /etc/init.d/oscars-wbui *option* * /etc/init.d/oscars-authN *option* * /etc/init.d/oscars-pce-bandwidth *option* * /etc/init.d/oscars-pce-vlan *option* * /etc/init.d/oscars-wsnbroker *option* * /etc/init.d/oscars-authZ *option* * /etc/init.d/oscars-pce-connectivity *option* * /etc/init.d/oscars-pss-stub *option* * /etc/init.d/oscars-coordinator *option* * /etc/init.d/oscars-pce-dijkstra *option*   Here *option is one of “start”, “stop”, “restart”, “status” or “help”* |

# Configuration and customization

This section describes the tools and methods necessary to configure an OSCARS 0.6 deployment.

## Create the first WBUI user

A user account is necessary to login to the OSCARS web page (WBUI). This step will also help you verify your installation. An initial administrative user needs to be created using a command-line script. Additional users can be created and other management activities can be perfomed from the WBUI once you login to the WBUI. To create an initial user, follow these commands:

|  |
| --- |
| cd $OSCARS\_DIST/tools  ./bin/idc-useradd |

You will be prompted for necessary values.

You will be prompted to enter the name of the organization the user you created is associated with. If you see no entries for organizations, you may need to add your first local organization by entering “1” when asked to choose the organization number.

Be sure to grant your user the OSCARS-administrator attribute (1) too. If you wish to create, query or list reservations from the WBUI, you should also give this user the OSCARS-engineer attribute.

Once you have reached this point, you have completed the basic set of steps required to check if your basic OSCARS installation is successful.

## Verifying that your basic installation was successful

The Web User Interface (WBUI) is a set of web pages provided with OSCARS for managing reservations and users. To initially verify that OSCARS installed correctly with the default settings, you can follow the WBUI URL and perform a few basic activities.

First, verify that the authN, authZ and wbui services are running. See section [Starting/Stopping OSCARS](#_Starting/Stopping_OSCARS) to start/stop services.

Then follow the below URL to use the WBUI.

<https://your-machine:8443/OSCARS>

Once there you will need to login as user you just created in section [Create the first WBUI user](#_Create_the_first).

You can use the browser interface to manager users, create, query or list reservations. Follow [URL](http://code.google.com/p/oscars-idc/wiki/Oscars_06_Use_Cases) to view a list of items that can help you test if your default installation went correctly.

## Define your local domain

You must define your local domain so OSCARS knows what elements to control. Every domain has a unique identifier. For example, the default domain that OSCARS is installed with is “testdomain-1”, Internet2’s domain ID is ion.internet2.edu and ESnet is identified by es.net. You are free to choose your own domain identifier.

The local domain used by your IDC installation can be viewed set using the steps below:

|  |
| --- |
| ./idc-localdomainview *context* |

You can change the local domain used by your IDC installation using the steps below:

|  |
| --- |
| cd $OSCARS\_DIST/tools/bin  ./idc-localdomainmod *context*  Here *context* is PRODUCTION or PRO |

You may manually change the local domain id in this file:

$OSCARS\_HOME/Utils/conf/config.yaml.

Set MPLS choice to 0 or 1.

## Define your Topology

### Creating an XML Topology Description

OSCARS currently requires you to manually generate an XML file that describes your network’s topology in the Open Grid Forum (OGF) Network Measurement Working Group (NMWG) control plane topology. The topology description you generate describes what is possible on your network. For example, it is not concerned with what VLANs are currently provisioned on a network, rather the possible VLANs that could be provisioned on the network.

#### Example XML files

The easiest way to generate this file is to start from the two examples provided with the OSCARS Software Suite. These files describe the same topology of a “testdomain-1” domain with 2 nodes, and a “testdomain-2” domain with 4 nodes. You can find the example XML files in the following locations:

$OSCARS\_HOME/TopoBridgeService/conf/testdomain-1.xml

$OSCARS\_HOME/TopoBridgeService/conf/testdomain-2.xml

### Configure Topology information

Once you have your topology files ready, you may use the following set of details to configure topology related settings.

#### Setting location of topology file

Once you have the topology file ready, you can place it in any location you find convenient, and instruct OSCARS to find it using the following commands:

|  |
| --- |
| cd $OSCARS\_DIST/tools/bin  ./idc-domaininfoadd *context*  Here *context* is PRODUCTION or PRO |

You will be prompted to enter your domain first. If the domain is not already present, you will then be prompted to specify whether it is a “file” or “topology server”. Choose “file”, since you have created a “topology file” for your local domain. You will then be prompted to enter the topology file name. The location can be either in the form of an absolute path, or a relative path. A relative path indicates that the Topology service will try to locate the file relative to $OSCARS\_HOME/TopoBridgeService/conf.

You may manually change this location by modifying file $OSCARS\_HOME/TopoBridgeService/conf/config.SSL.yaml. Specify the *source* as 'file', and key in your domain for the *file* field.

#### Adding topology server information

OSCARS provides a Topology Bridge Service whose function is also to store topology information. This topology information can be stored or located in two ways, currently. The first method is that of storing the topology information in a static file, and instructing OSCARS to look for the topology for a particular domain in a static file. The second method is to indicate OSCARS to look for a particular domain via a bridge to a PerfSONAR topology Server.

You can add information about such a topology server using the following commands.

|  |
| --- |
| cd $OSCARS\_DIST/tools/bin  ./idc-toposerveradd *context*  Here *context* is PRODUCTION or PRO |

You will be now be prompted to enter the domain. If you would not want to specify a particular domain, and would like to use a PerfSonar Topology Server for all domains you did not specify individually, then choose “\*” when you are prompted for a domain.

If the domain is not already present, the Topology Server details will be added afresh. On the other hand, if the domain is already present, the topology Server location you specify will be added on to this. In other words, the same script can be used to add and modify your topology server information. Also, having more than one Topology Server for a specific domain simply means that OSCARS will try to locate the topology associated with this domain using these Topology Servers in succession.

You may manually change this location by modifying file $OSCARS\_HOME/TopoBridgeService/conf/config.SSL.yaml. Specify the *source* as ‘topoServer’, and key in your Topology Server details for the *servers* field. More than one server can be added ; just separate the entries by a comma. For example:

|  |
| --- |
| '\*':  source: 'topoServer'  servers: [ 'ts-server.domain.org' , ‘ts-2.edu’] |

#### Modifying Domain Information

The previous two sections provided details about adding topology information to OSCARS.

However, you may also want to modify your current configuration and associate a different file or with this domain. The commands below will help you achieve this.

|  |
| --- |
| cd $OSCARS\_DIST/tools/bin  ./idc-domaininfomod *context*  Here *context* is PRODUCTION or PRO |

You will be prompted to enter a domain. Once you have entered a domain name, you will be prompted to specify whether it is a File/Topology Server you would like to newly associate to this domain. Note that you may even change your original setup to anything among File or TopologyServer .In other words, if you had originally associated domain X with a “file”, you can switch it to use a Topology Server instead).

You may manually change this location by modifying file $OSCARS\_HOME/TopoBridgeService/conf/config.SSL.yaml. If you specify the *source* as ‘file’, then record your file location in the *file* field. For example:

|  |
| --- |
| 'testdomain-4':  source: 'file'  file: 'testdomain-4.xml' |
| The location can be either in the form of an absolute path or a relative path. A relative path indicates that the Topology service will try to locate the file relative to $OSCARS\_HOME/TopoBridgeService/conf. |

If you specify *source* as ‘topoServer’ , then record your Topology Server details for the *servers* field. More than one server can be added ; just separate the entries by a comma. For example:

|  |
| --- |
| '\*':  source: 'topoServer'  servers: [ 'ts-server.domain.org' , ‘ts-2.edu’] |

## Changing ports used

You can customize the ports on which OSCARS services are being run currently.

To view the ports currently being used, run the following set of commands:

|  |
| --- |
| cd $OSCARS\_DIST/tools/bin  ./idc-portview context server |

Here context is PRODUCTION or PRO. You can view the port number used by any specific service by using a specific option for “server”, or specifying “ALL”. You can see all servers listed if you specify none.

To change any of the ports being used by the different services, use the following set of commands:

|  |
| --- |
| cd $OSCARS\_DIST/tools/bin  ./idc-portmod context server |

The current ports being used are:

api: 9001, 9002 (internal)

authN: 9090

authNPolicy: 9004

authNStub: 9011

authZ: 9190

authZPolicy: 9005

coordinator: 9003

coordinator:pceRuntime: 10000

lookup: 9014

notify: 9012

PCEs

stubPCE 9007 (only run if no other PCE’s are run)

connectivityPCE 9007

bandwidthPCE 9009

vlanPCE 9010

dijkstraPCE: 9008

resourceManager: 9006

PSS: 9050

topoBridge: 9019

wbui:http 8080

wbui:https 8443

ionui: http 9195

ionui: http 9196

Alternately, you may change the ports used by manually editing lines like:

<httpj:engine port="9019"> in files

$OSCARS\_HOME/<servicename>/conf/server-cxf-ssl.xml

## Changing your DB password

It is recommended that the OSCARS MySql password for the “OSCARS” user be modified before setting in critical user/reservation information. To do so, follow these steps:

|  |
| --- |
| cd $OSCARS\_DIST/tools/bin  ./ idc-dbpassmod |

This script offers 3 options:

1. Change Database password in ALL locations (Default option)

2. Change OSCARS references to Database password

3. Change MySQL Database password

Selecting option 1 will to change both the MySQL and OSCARS configuration file references to your intended values. This option is the default option. You will be prompted for your old password, which will be used to authenticate and then change over to the new value you provide.

Option 2 is used to change the OSCARS password in configuration files referenced by OSCARS. For example, if you intend to manually change the MySql password, OSCARS can then be made to use the same by running the script with this option.

Option 3 is a utility which helps set only the DB password. Please note that this is for the “Oscars” MySQL user alone.

Alternatively, you may modify the password in files

$OSCARS\_HOME/AuthNService/conf/authN.SSL.yaml

$OSCARS\_HOME /AuthZService/conf/authZ.SSL.yaml

$OSCARS\_HOME /ResourceManagerService/conf/config.SSL.yaml

$OSCARS\_HOME/ IONUIService/conf/config.SSL.yaml

#### Key stores

OSCARS creates a set of default “test” keys that are used by the various services and in inter-domain communication.

The RPM installation sets up all services to use two key stores:

|  |  |
| --- | --- |
| **Key store** | **Purpose** |
| $OSCARS\_HOME/keystores/oscars.jks | The key in this key-store is used by the WSNBrokerService for HTTPS, and by the OSCARSService to sign messages. This key-store also acts as the trust-store for keys used by all other services, and must contain the Certifying authority (CA) chain of any key used by other services. |
| $OSCARS\_HOME /keystores/localhost.jks | The key contained by this key-store is used by all other OSCARS services for HTTPS. Since all local services also need to communicate with the WSNBrokerService/OSCARSService, this must contain the Certifying authority (CA) chain of any key used by other services, i.e the CAs of the keys in oscars.jks. |

To use your own set of keys, your may follow these generic steps:

* 1. Generate a generate a new certificate and private key for your IDC
  2. Create a certificate signing request (CSR) and pass your request to a Certifying authority (CA) who will verify your credentials and return a signed certificate to you.
  3. Import this certificate into $OSCARS\_HOME/keystore/oscars.jks
  4. Import the CA chain of your key into $OSCARS\_HOME/keystore/localhost.jks
  5. Create a new key to be used by your local services
  6. Import this key (may be signed again by a CA) into localhost.jks, and its CA chain into $OSCARS\_HOME/keystore/oscars.jks
  7. To copy certificates, use *keytool -export* and then -*import*.

For more information, refer to

<http://code.google.com/p/oscars-idc/wiki/KeystoreFiles>

#### Change references to certificate alias

Change certificate alias from 'mykey' into the alias you used when creating new keys in oscars.jks. This applies to configuration files in the WSNBrokerService and OSCARSService folders.

* + Open WSNBrokerService/conf/client-cxf-http.xml
  + Find lines like

<entry key="ws-security.signature.username" value="mykey" />

* + Replace “mykey” with the certificate alias from rampConfig.xml
  + Repeat this procedure in files:

|  |
| --- |
| $OSCARS\_HOME/OSCARSService/conf/client-cxf-ssl.xml  $OSCARS\_HOME /OSCARSService/conf/client-ut-cxf-ssl.xml  $OSCARS\_HOME /OSCARSService/conf/server-cxf-ssl.xml  $OSCARS\_HOME /OSCARSService/conf/client-cxf-DEV.xml  $OSCARS\_HOME /WSNBrokerService/conf/client-cxf-ssl.xml  $OSCARS\_HOME /WSNBrokerService/conf/server-cxf-ssl.xml |

Change certificate alias from 'mykey' into the alias you used when creating new keys in localhost.jks. This applies to configuration files in the NotificationBridgeService folders.

|  |
| --- |
| $OSCARS\_HOME /NotificationBridgeService/conf/client-cxf-ssl.xml |

# Creating and Managing User Accounts

The OSCARS IDC has a built-in system for authenticating and authorizing requests. User information is kept in the MySQL database. The WBUI associates a username and password with accounts kept in this database. Creating user accounts can be done via the WBUI as described in section **6.1**.

## Adding Users to the Database

Users and permissions are stored in a MySQL database. This section details how to add users to the database and assign them permissions.

### Creating New User Accounts

You can create new users from the OSCARS web page called the web user interface (WBUI). The five steps for creating a new user via this method are:

1. Visit https://[your-server:8443/OSCARS](http://your-server:8080/OSCARS) – where *your-server* is the name of the server on which the WBUI is running
2. Login with a user account that has administrator privileges (such as the one you created in [Create the first WBUI user](#_Create_the_first_1))
3. Click the **Users** tab on the top of the page that loads
4. Click the **Add User** button
5. Complete all the fields outlined in green. Most of the fields are self-explanatory but a few are worth mentioning:
   * *X.509 Subject Name* – Use this field to associate an X.509 certificate with a particular user. This field is not required to allow the user to provision via the WBUI. It is required if the entity associated with this account will send requests using the web service API (such as another IDC. TBD.
   * *X.509 issuer name –* Generally the X.509 Subject Name is enough but you may also use this field to indicate the issuer of the certificate. In most cases you do not need to use this field.
   * *Choose Role(s) –* These determine what permissions users have. More complicated permissions are beyond the scope of this document.
6. Click the **Add** button on the top of the screen

### Modifying Users

You may modify user accounts via the WBUI under the **Users** tab. Clicking on the users last name will display a form for editing the user’s profile. The form should be self-explanatory as it is similar to the add users form.

### Deleting Users

You may delete user accounts via the WBUI under the Users tab. Clicking on DELETE will remove the user after a confirmation

## Managing X.509 Certificates from Users and Other IDCs

TBD

# Using the ION UI

If you wish to use the ION UI web interface to use request reservations and use other OSCARS service capabilities, you could install it using command:

|  |
| --- |
| yum install oscars-ionui |

To be aboe to log in initially and create other users, you must first create an administrative user. To do this, run steps:

|  |
| --- |
| cd $OSCARS\_DIST/tools  ./bin/idc-useradd |

The idc-useradd command will prompt for all the necessary values. Be sure to grant your user the

ION-administrator. Once you have created this user, you can log in to the IONUI and modify your attributes or create new user users and assign various attributes to them.

## Starting/Stopping ION UI Service

You can start the IONUI server by using this command:

|  |
| --- |
| /etc/init.d/oscars-ionui options  where *options* are one among start, stop, restart or help |

Please note that currently, the “ALL” option that can be used with /etc/init.d/oscars (to start all servers) does not include the ION UI service. You will thus need to run the above start command to individually start ION.

Based on the context you choose, you will see an output print indicating the port on which the server has been started. Currently, IONUI runs on ports 9195, or 9196 if using SSL (i.e. https). Also currently, the “production”, and “development” context uses SSL, while the other 2 do not.

Verify that the authN , authZ and ionui servers are running. If you chose the context to be “production” or “development, you can access the ION UI using URL

[https://yourhost.yourdomain:9196/ion/](https://localhost:9196/ion/)

Once there you will need to login as user you just created. Currently you can use the browser interface to create, query or list reservations and manage users.

## Configuring the ION UI Service

### PerfSonar-PS components

To be able to effectively use the ION UI, you need some PerfSonar components. These are the lookup service (LS), the topology service (TS) and the friendly names service. These enable the user to associate “friendly names” to endpoint links in your topology, and makes creating circuits easier. To install these, use the instructions [here](http://www.internet2.edu/ion/install.html). Refer to the “Deploying the Lookup Service”, “Deploying the Topology Service” and “Deploying the Friendly Names Service” sections.

Once you have completed these steps, you can then configure your ION installation to use these.

#### Endpoint Browser Configuration

Endpoints can be associated with friendly names that make them easier to identify than raw link URLs.  
The endpoint browser uses the data dumps of friendly-names from the LS. The perfSONAR\_PS-DCNNameAdmin packages include a script dcn\_dump that outputs the friendly names registered in the lookup service. This dump file needs to be made available via HTTP. Since this package is already installed on the host with the LS, it's probably easiest to add a cron entry to dump the .csv file to a location where it's accessible via HTTP. You could modify scripts/regular\_backup.sh to make a copy to some easily accessible location.

Once you have that at a given URL, change the URL in the “endpoints” section of '$OSCARS\_HOME/IONUIService/conf/config.XXX.yaml’ file to the new URL. Here XXX is either SSL or HTTP.

Then change the 'endpointData.localDomain' string to the local domain you have set up on your installation. This is the same local domain your topology files/other configuration files use.

For example, in **$OSCARS\_HOME/IONUIService/conf/config.SSL.yaml**,

the endpoints section could be:

endpoint:

endpointData.url:'http://yourhost.yourdomain/friendly\_names.csv'

endpointData.localDomain:'testdomain-1'

#### Topology Configuration

The file $OSCARS\_HOME/IONUIService/conf/ion\_topology.yaml' needs to be configured. This configuration is used for two aspects: the gps coordinates and ingress/egress utilization.

Check the “Topology configuration” section [here](http://www.internet2.edu/ion/install.html) for examples on configuring this file.

After you have completed these steps, restart your server using the commands in the “Starting/Stopping ION UI Service” section above.

### Database (MySql) password

As mentioned earlier, anyone who has login access to the host on which the mysql server is running and knows the oscars mysql password can modify the OSCARS database files with mysql commands. Thus you should change the default password and protect the files that contain it to the user that the services will run as. The distribution includes the following ION related files containing the password:

ionui/config/config.HTTP.yaml where XXX=SSL or HTTP

The deployOscarsSrc.sh script will copy the yaml.template files to .yaml files if such files do not already exist. To change the default password edit the \*.yaml files and change the protection of those files to be readable only by the userId that the service runs under.

Alternatively, you could use these commands to set OSCARS MySql password:

|  |
| --- |
| cd $OSCARS\_DIST/tools  bin/idc-dbpassmod |

You will be prompted to choose options to change the database password and all references to the same in the OSCARS files, or just the DB password, or just the OSCARS references.

Note that the password is common across all other services like AuthN, AuthZ and Resourcemanager that use MySQL databases too.

### Port number

If you wish to change the ports on which you run IONUI, use these commands:

|  |
| --- |
| cd $OSCARS\_DIST/tools  bin/idc-portmod *context* ionui  Here *context* is PRODUCTION or PRO |

You will now be prompted to enter a new port, and can proceed to use one of your choice. You can then restart your IONUI server using the commands in the “Starting/Accessing ION UI Service” section above. From this point onwards, you can access ion using an URL indicating your new choice of port (for example, https://yourhost.yourdomain:4444/ion/).

# Inter-domain configuration

This section details the steps required for configuring your IDC to communicate with other IDCs.

## Making your IDC Aware of Other Domains

Make your 0.6 domain aware of another 0.6 domain you want to peer this way:

|  |
| --- |
| cd $OSCARS\_DIST/lookup  ./bin/oscars-idcadd -d *peer\_domain* -p http://oscars.es.net/OSCARS/06 -l http://*peer\_host.peer\_domain*:9001/OSCARS |

Here

* + peer\_domain is the neighbour domain you wish to peer with
  + the input to the **–l** option is the URL of the peer installations’ “OSCARSService” URL. If no ports are changed at the peer’s end, just changing the *peer\_host.peer\_domain* values should suffice
  + the input to the **–p** option is a constant

Remember to run this step on both the 0.6 domains that you want to peer.

## IDC certificates for sending inter-domain requests

TBD

## Test your sample multi-domain installation

OSCARS some installed with some sample files (topology, key stores) that may help you run a test to check your multi-domain installation.

Do to so, you will need two OSCARS IDC services running. The first should have a localDomain:id in $OSCARS\_HOME/TopoBridgeService/conf/config.SSL.yaml set to testdomain-3, the second should have testdomain-4 as its localDomain:id. An OSCARSService finds a peer service for another domain using a Lookup Service. The lookup service that we distribute is a bridge server can contact an external Lookup service and cache the URLs of services for peer domains. Since we are not currently deploying and external service, we need to add the service/domain information directly to the Lookup Service cache. This is done for testdomain-3 (running on host3) and testdomain-4 (running on host3) by the following commands:

On the host3 that is serving testdomain-3:

cd $OSCARS\_DIST/lookup/

bin/oscars-idcadd -d testdomain-4 -p <http://oscars.es.net/OSCARS/06>

-l [http://host4:9001/OSCARS](http://odev-vm-12.es.net:9001/OSCARS)

On the host4 that is serving testdomain-4:

cd $OSCARS\_DIST/lookup/

bin/oscars-idcadd -d testdomain-3 -p <http://oscars.es.net/OSCARS/06>

-l [http://host3:9001/OSCARS](http://odev-vm-12.es.net:9001/OSCARS)

These commands will add the appropriate entries to $OSCARS\_HOME/LookupService/data and only need to be repeated if you delete that directory.

The two servers also need to share a common root CA and RA, so that the signed messages that are passed between them can be verified. To accomplish this:

On the primary IDC domain, e.g testdomain-3, when first running sampledomain/bin/gencerts use an empty $OSCARS\_HOME/sampleDomain/certs which will cause a new CA, RA, client, localhost and oscarsidc keystores to be created. On the second IDC, testdomain-4, copy the sharedCred.tar file that was created in the first domain into $OSCARS\_HOME/sampleDomain/certs on the second host. Then this command will use the shared RA to generate and sign a new oscarsidc certificate and keystore.

# Customizing logging for your deployment

Getting the right amount of logging is often a delicate balance, so we have provided ways to customize the logging in the OSCARS configuration files. In our default logging setup all the log4j messages (root logging) are put into per service .out files in $OSCARS\_DIST. In addition, only the OSCARS specific messages are put into per service log files in $OSCARS\_HOME/logs. We have provided three default logging profiles:

INFO: intended for production systems that only logs *info* level log messages and in addition puts the log messages from all the services into $OSCARS\_HOME/oscars.log

DEBUG: which logs the *debug* level log messages

MESSAGES: which logs *debug* level log messages and enables the cxf message logging which will put the all the inter-service messages in the .out files.

By default, the PRODUCTION context uses the INFO log files. Note that the message logging can be very verbose for the coordinator/pce messages as they contain the entire topography. One can change the log level used by any service by editing that service’s manifest file.

To customize, increase or reduce message logging by editing the manifest.yaml file.

For example,

* + Open file $OSCARS\_HOME/WSNBrokerService/conf/manifest.yaml
  + Locate section under header “PRODUCTION”
  + Locate the “log4j.properties” file. This may be something like 'OSCARS\_HOME/WSNBrokerService/conf/log4j.INFO.properties'
  + Change this to point to log4j.DEBUG.properties if you want to increase the log level.