
**ARCHER 1.0**

# ARCHER Data Services Service Layer

## System Administrator's Guide

### ICAT & MCAText

- Installation
- Configuration
- Maintenance

## About ADS Service Layer

### Overview

ARCHER Data Services (ADS) Service Layer is composed of two web applications, ICAT and MCAText.

**ICAT** is a metadata storage service that implements the CCLRC Scientific Metadata Model version 2 to record information about scientific experiments. The data from the experiments itself is stored on the SRB, while the metadata is held in the ICAT. The ICAT's storage is implemented as a PostgreSQL database, which is installed through the Archer XDMS application.

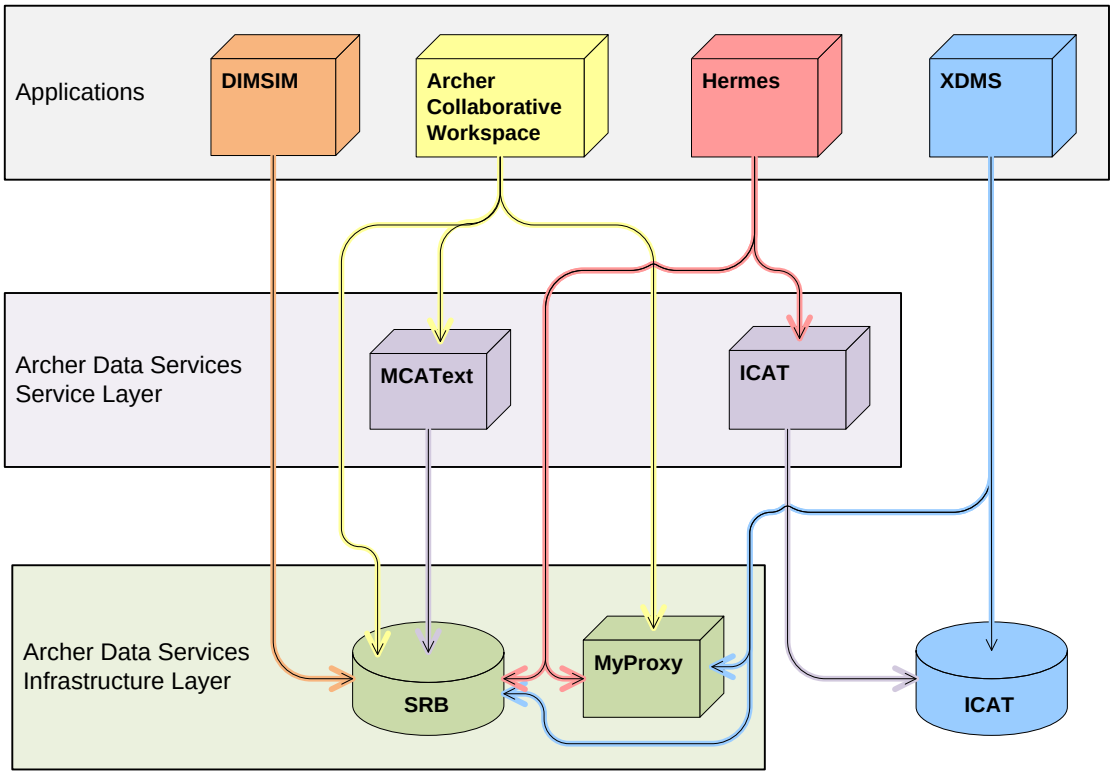
**MCAText** is an ARCHER-developed web service layer over SRB and its MCAT database. It provides a high performance mechanism for other services to lookup authorisation information on content within SRB. It provides update notification to other services when content is modified, moved, or created. It is used by certain ARCHER tools, including the ICAT service and ARCHER Collaborative Workspace.

You must install the ADS Infrastructure Layer, including SRB and MyProxy, before installing ADS Service Layer.

### Do I need this?

ADS-SL is used as follows:

- ARCHER's **Hermes** communicates with the ICAT service to browse experiments.
- **ARCHER Collaborative Workspace** (Plone) communicates with MCAText to browse the SRB.
- ARCHER development and testing identified that a future version of XDMS could use the ICAT service rather than accessing the ICAT database directly.




**Applications depending on ADS-SL**

## Architecture

ICAT consists of a web application and the ICAT PostgreSQL database created by XDMS.

MCAText consists of a web application which uses the MCAT database already created as part of SRB.

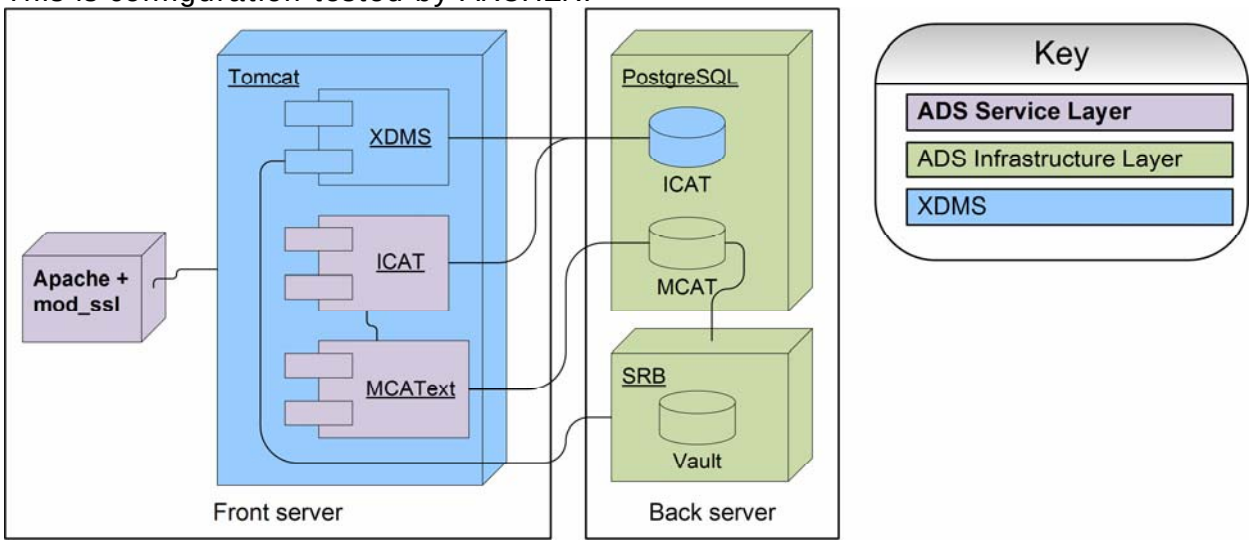
Both web applications are hosted by Tomcat, and are generally accessed through an Apache server.

In the standard configuration that was tested by the ARCHER project:

- The XDMS, ICAT, and MCAText web applications are hosted by the same Tomcat.
- The ICAT and MCAT databases are hosted by the same PostgreSQL.
- Tomcat and Apache are on the same “front” server.
- PostgreSQL and SRB are on the same “back” server.

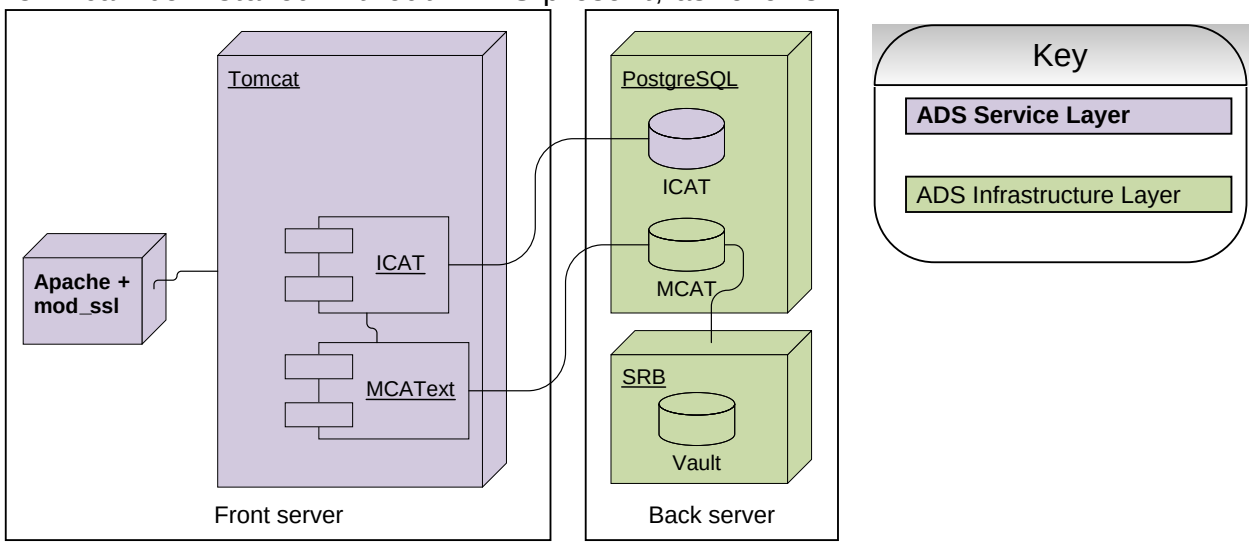
**With XDMS:**

This is configuration tested by ARCHER.




**Without XDMS:**

ICAT can be installed without XDMS present, as follows:




When installing ICAT without XDMS, there is an additional database creation step described below. Note that this configuration has not been tested by ARCHER.

## Dependencies

These ARCHER components must be installed first:

| ARCHER component | Creates | Reason required |
|---|---|---|
| ADS Infrastructure Layer | MCAT database (PostgreSQL)<br><br>SRB | Provides back end to MCAText web service layer. |
| | CA (optional) | Used to generate certificates which are used in this installation process. |
| | MyProxy server | Used to give MCAText access to SRB. |

| | | |
|---|---|---|
| XDMS (semi-optional) | ICAT database (PostgreSQL) | Provides back end to ICAT web service layer. |
| | Tomcat application server | Hosts ICAT and MCAText web applications. |

These components are also required:

- Apache web server 2.2 or later, with `mod_ssl`.

Typically, the Apache server is on the same machine as Tomcat, but need not be.

**If you have not already installed Apache web server:**

```
yum install httpd mod_ssl
```

To install Subversion:

```
yum install subversion
```

## Non-standard configurations

### ICAT and MCAText separate from XDMS

It is not strictly necessary that ICAT and MCAText be deployed in the same Tomcat container as XDMS. However, due to the shared libraries used by the three web applications, hosting them on the same machine is a more efficient use of memory.

To install ADS SL on a separate server from XDMS, you must install another instance of Tomcat. Obtain Apache Tomcat version 5.5 from `http://tomcat.apache.org/download-55.cgi`.

Install Tomcat to `/usr/local/archer/tomcat` and run it as a user called `tomcat`.

Then download the PostgreSQL JDBC driver and place it in `/common/libs` of your Tomcat installation. This driver is found at `http://jdbc.postgresql.org/`.

**Note:** Installing Tomcat through Yum is not recommended. Difficulties were encountered by the ARCHER project.

### ICAT and MCAText separate from each other

It is also not strictly necessary that ICAT and MCAText be deployed in the same Tomcat container as each other. However, to arrange this will require that the installation be carried out twice, with some manual configuration. This method is not described here, as there is no particular benefit to doing this.

## Installing ICAT and MCAText

### Overview

ICAT and MCAText are installed and configured simultaneously.

The major steps are as follows:

1. Obtain the configuration scripts and web service packages.
2. Obtain or generate certificates.
3. (If required) Create the ICAT database.
4. Add PL/pgSQL to the MCAT database
5. Set environment variables for configuration.
6. Run the script to generate deployment files.
7. Deploy ICAT and MCAText.
8. Install and configure Apache.

### 1. Obtaining ICAT and MCAText

Download the ADS-SL bundle from http://www.archer.edu.au/downloads.

As the `tomcat` user, unzip it to a permanent location. This document assumes `/usr/local/archer/icat_mcatext` .

```
# mkdir -p /usr/local/archer/icat_mcatext
# chown tomcat /usr/local/archer/icat_mcatext
# su tomcat

$ wget http://www.archer.edu.au/downloads/ads-sl-1.0.tar.gz
$ tar –xzf ads-sl-1.0.tar.gz –C /usr/local/archer/
```

The distribution contains the following files:

| File | Purpose |
|---|---|
| `icat.war` | Web archive file for ICAT webservice. |
| `mcatext.war` | Web archive file for MCAText webservice. |
| `install.sh` | Script you will run to configure ICAT and MCAText. |
| `AddCertToKeystore.class` | Used by install script to add certificates to a Java keystore (JKS). |
| `AddCertToKeystore.java` | Source file. Not used in installation. |
| `makekeystore.sh` | Used by install script to create keystore. |
| `xdms_icat_ddl.sql` | Script to create the ICAT database, if XDMS is not present. |
| `xdms_icat_dml.sql` | Script to populate the ICAT database, if XDMS is not present. |
| `templates/` | Template context files for Tomcat, used by the install script. |

### 2. Obtaining or creating certificates and keys

A total of four host certificate/key pairs are required: ICAT, MCAText, the server itself, and Apache. You can use the same certificate/key pair for the server and Apache. Using three separate pairs assists in fine-grained security control.

This document assumes the same certificate/key pair will be used for the server itself and Apache.

**If you are using the ARCHER MyProxy scripts as a CA:**

On the CA machine, run `cert_tool`[1] as follows:

```
cert_tool -s -c icat@server.uni.edu.au -e admin@uni.edu.au
cert_tool -s -c mcatext@server.uni.edu.au -e admin@uni.edu.au
cert_tool –s –c server.uni.edu.au –e admin@uni.edu.au
```

In place of `server.uni.edu.au`, use the fully-qualified domain name of the ICAT host machine.

The files are generated in a `/tmp` directory, which is printed out by the tool. The CA certificate file is already present in `/etc/grid-security/certificates`, with a name like `fd7ecfa4.0` .

**If you are using a different CA:**

You must obtain three certificate and keys as follows, plus the CA certificate:

1) Host certificate/key for ICAT.
   Common Name: `icat@server.uni.edu.au`
2) Host certificate/key for MCAText.
   Common Name: `mcatext@server.uni.edu.au`
3) Host certificate/key for server itself.
   Common name: `server.uni.edu.au`
4) CA certificate itself.

**Note:** It is possible to use just one host key/certificate for all services. In this case, you would use a common name like `server.uni.edu.au` instead.

Certificate and key files must be provided in .pem format. If you receive them in a different format, you must convert them first.

Copy these files to the same directory as the installation scripts. Rename them as follows:

| Key/certificate | Rename as… | Copy to… |
|---|---|---|
| ICAT server certificate | `icatcert.pem` | Install directory. |
| ICAT host keys | `icatkey.pem` | |
| MCAText host certificate | `mcatextcert.pem` | |
| MCAText host keys | `mcatextkey.pem` | |
| Certificate for CA itself. | `cacert.pem` | |
| Host certificate | `hostcert.pem`<br>`httpdcert.pem` | `/etc/grid-security` on Apache server machine. |
| Host key | `hostkey.pem`<br>`httpdkey.pem` | |

[1] For documentation on `cert_tool`, see the ADS Infrastructure Layer System Administrator's Guide. `cert_tool` is installed in `/usr/local/sbin`.

Ensure that all files have appropriate permissions:

- Key files must not be group or world readable (`chmod 600`)
- Certificate files must be world readable (`chmod 644`)
- Apache certificate and key (`httpdcert.pem` and `httpdkey.pem`) must be owned by apache

For example, assuming certificates provided as `icat_certs.tgz`, `mcatext_certs.tgz`, and `host_certs.tgz` in your home directory:

```
cd /usr/local/archer/icat_mcatext

tar –zxf ~/icat_certs.tgz hostcert.pem > icatcert.pem
tar –zxf ~/icat_certs.tgz hostkey.pem > icatkey.pem

tar –zxf ~/mcatext_certs.tgz hostcert.pem > mcatextcert.pem
tar –zxf ~/mcatext_certs.tgz hostkey.pem > mcatextkey.pem

chmod 600 *key.pem
chmod 644 *cert.pem

# Assuming Apache is on this machine:

cd /etc/grid-security
tar –zxf ~/host_certs.tgz hostcert.pem > hostcert.pem
tar –zxf ~/host_certs.tgz hostkey.pem > hostkey.pem

cp hostcert.pem httpdcert.pem
cp hostkey.pem  httpdkey.pem

chmod 600 *key.pem
chmod 644 *cert.pem

chown apache httpd*.pem

ls -l /etc/grid-security/*.pem /usr/local/archer/icat_mcatext/*.pem
```

```
-rw-r--r-- 1 root   root   /etc/grid-security/hostcert.pem
-rw------- 1 root   root   /etc/grid-security/hostkey.pem
-rw-r--r-- 1 apache root   /etc/grid-security/httpdcert.pem
-rw------- 1 apache root   /etc/grid-security/httpdkey.pem
-rw-r--r-- 1 root   root   /etc/grid-security/req.pem
-rw-r--r-- 1 root   root   /usr/local/archer/icat_mcatext/cacert.pem
-rw-r--r-- 1 root   root   /usr/local/archer/icat_mcatext/icatcert.pem
-rw------- 1 root   root   /usr/local/archer/icat_mcatext/icatkey.pem
-rw-r--r-- 1 root   root   /usr/local/archer/icat_mcatext/mcatextcert.pem
-rw------- 1 root   root   /usr/local/archer/icat_mcatext/mcatextkey.pem
-rw-r--r-- 1 root   root   /usr/local/archer/icat_mcatext/req.pem
```

## 3. (Optional) Creating ICAT database

**If you have XDMS installed, skip to step 4.**
The ARCHER project tested ICAT installed using the same database as XDMS. However, it is theoretically possible, though untested, to install ICAT without XDMS.

Two SQL scripts are required:

- `xdms_icat_ddl.sql` creates the ICAT table structure.
- `xdms_icat_dml.sql` populates it with some default values.

These files are included in the ICAT source bundle. You should edit `xdms_icat_dml.sql`, tweaking the values for your needs.

On the database machine:

| Step | Typical command |
|---|---|
| 1. Install PostgreSQL, if not already present. | `yum install postgresql` |
| 2. Switch to postgres user. | `su – postgres` |
| 3. Create a user called 'icat'. | `createuser icat --pwprompt --no-superuser --no-createdb --no-createrole` |
| 4. Create a database called 'icat'. | `createdb icat --owner icat` |
| 5. Run the DDL script to create the ICAT database structure. | `psql –dbname icat --file xdms_icat_ddl.sql --username icat` |
| 6. Run the DML script to populate the ICAT database structure. | `psql –dbname icat --file xdms_icat_dml.sql --username icat` |

**Note:** The ICAT user must have read and write access to all ICAT tables. If using a different method to create the database and tables, you can grant access with this SQL command:

```
GRANT ALL PRIVILEGES ON DATABASE icat to icat;
```

## 4. Adding PL/pgSQL to MCAT

MCAText requires the PL/pgSQL language for stored procedures to be enabled in the MCAT database. MCAT is SRB's metadata database and was installed with SRB.

On the machine hosting MCAT, run these commands:

```
# su - postgres

$ createlang plpgsql MCAT
```

You can verify that this worked as follows.

```
$ createlang -l MCAT

Procedural Languages
  Name   | Trusted?
---------+----------
 plpgsql | yes
```

## 5. Set environment variables

The install script uses a number of environment variables. If certificates and `.war` files are located as described in this document, many of the default values can be used.

Check the defaults in the table below, and set any variables as needed.

In particular you must set the name of the SRB host, and passwords for the two databases. For example:

```
export SRB_HOSTNAME=srb.uni.edu.au
export ICAT_DB_PASSWORD=xxxx
export MCATEXT_DB_PASSWORD=xxxx
```

| Variable | Contains | Defaults to |
|---|---|---|
| CATALINA_HOME | Location of Tomcat | |
| SRB_HOSTNAME | Host name of SRB server | |

| | | |
|---|---|---|
| XDMS_BASEPATH | SRB URL to XDMS project area. For example: srb://srbhost/myzone/home/xdms_project | |
| ICAT_CLIENT_CERT | Path to ICAT host certificate file | ./icatcert.pem[2] |
| ICAT_CLIENT_KEY | Path to ICAT host key file | ./icatkey.pem |
| MCATEXT_CLIENT_CERT | Path to MCAText host certificate file | ./mcatextcert.pem |
| MCATEXT_CLIENT_KEY | Path to MCAText host key file | ./mcatextkey.pem |
| CA_CERT | Path to CA certificate file | ./cacert.pem |
| ICAT_WAR | Path to ICAT .war file | ./icat-webservice-1.0.war |
| MCATEXT_WAR | Path to MCAText .war file | ./mcatext-webservice-1.0.war |
| ICAT_DB_HOSTNAME | Host of PostgreSQL for ICAT | localhost |
| ICAT_DB_DBNAME | Name of ICAT database | icat |
| ICAT_DB_USERNAME | Username/password for ICAT database | xdms |
| ICAT_DB_PASSWORD | | |
| MCATEXT_DB_HOSTNAME | Host of PostgreSQL DB for MCAT | $SRB_HOSTNAME |
| MCATEXT_DB_DBNAME | Name of MCAT database | mcat |
| MCATEXT_DB_USERNAME | Username/password for MCAT database | srb |
| MCATEXT_DB_PASSWORD | | |

## 6. Running the configuration script

The configuration script uses the environment variables you have set to create two Tomcat context files, two Java keystores, and a whitelist for MCAText.

Run it as follows:

```
$ ./install.sh
```

If any required environment variables have not been set, you will be advised, and the script will stop.

The script generates these files in the current directory:

| Filename | Contains |
|---|---|
| icat.jks | Java keystore for ICAT, containing the provided keys and certificates. |
| mcatext.jks | Java keystore for MCAText, containing the provided keys and certificates. |
| mcatext-whitelist | Whitelist for MCAText, containing ICAT. This file tells MCAText which hosts to allow connections from. |
| icat.xml | Tomcat context file for ICAT |
| mcatext.xml | Tomcat context file for MCAText. |

Verify the contents of the Tomcat context files, `icat.xml` and `mcatext.xml`. Ensure that all variables have been substituted correctly.

If required, modify your variables, then re-run `install.sh`.

[2] The actual absolute current directory path is stored, rather than a relative path.

## 7. Deploying context files

Now that the context files have been generated, deploy them to Tomcat.

1. Stop Tomcat.
   ```
   # $CATALINA_HOME/bin/shutdown.sh
   ```

2. Copy `icat.xml` and `mcatext.xml` to
   ```
   # $CATALINA_HOME/conf/Catalina/localhost
   ```

   ```
   # cp icat.xml $CATALINA_HOME/conf/Catalina/localhost
   # cp mcatext.xml $CATALINA_HOME/conf/Catalina/localhost
   ```

3. If it has not already been done[3], copy the PostgreSQL JDBC to Tomcat's `common/libs` directory. For example:

   ```
   # cd $CATALINA_HOME/common/libs
   # wget http://jdbc.postgresql.org/download/postgresql-8.3-603.jdbc4.jar
   ```

4. Restart Tomcat.
   ```
   # $CATALINA_HOME/bin/startup.sh
   ```

**Note:** The context files point to the `.war` files in their current location. So, do not move these files, or update the context files if you do.

**Note:** Ensure that the `tomcat` user can read the `.xml` files.

### Verifying Tomcat deployment

By default, MCAText and ICAT are set to only accept authenticated connections, so you can't connect to them until Apache is configured. However, you can verify that they are running as follows.

1. Connect to the server using an address like:
   `http://localhost:8080/icat/ws`

   Adjust this address as appropriate.
2. Check for a message that reads:
   `org.acegisecurity.AccessDeniedException: Access is denied`

   This indicates that ICAT has started up, but is rejecting the request due to lack of authentication.
3. Repeat steps 1 and 2 for MCAText:
   `http://localhost:8080/mcatext/ws`

The Tomcat log file also shows the web services starting up. See the Maintenance section for details.

## 8. Configuring Apache SSL

Now that the keys and certificates are obtained, they need to be registered in Apache.

[3] If you have already installed XDMS on this Tomcat, then you have already performed this step.

Add six lines to the `/etc/httpd/conf.d/ssl.conf`, just prior to the `</VirtualHost>` line, as follows:

| Line | Purpose |
|---|---|
| `SSLCertificateFile /etc/grid-security/httpdcert.pem` | Points to the location of the host certificate. |
| `SSLCACertificateFile /etc/grid-security/certificates/1e271185.0` | Points to the location of the CA certificate. |
| `SSLCertificateKeyFile /etc/grid-security/httpdkey.pem` | Points to the location of the host key. |
| `SSLVerifyClient optional` | Allows client connections to present certificates for verification, but does not require it. ICAT and MCAText themselves require authentication, so if they are the only services on this machine, you may wish to use "required". |
| `SSLOptions +StdEnvVars` | Tells Apache to create environment variables. Required for the next line. |
| `RequestHeader add SSL_CLIENT_S_DN %{SSL_CLIENT_S_DN}e` | Tells Apache to add the distinguished name (DN) of the client to its HTTP headers. There are used by MCAText to determine authorisation. |

Check whether any of these variables were already defined in this file, and comment them out if so.

Then, add the following three lines after them. These define the external address of the ICAT and MCAText services.

```
…
       RewriteEngine on
       RewriteRule ^/mcatext/(.*) ajp://localhost:8009/mcatext/$1 [L,P]
       RewriteRule ^/icat/(.*) ajp://localhost:8009/icat/$1 [L,P]
</VirtualHost>
```

This allows Apache to serve the Tomcat servlet. Add the correct server name for the Tomcat machine.

Then start Apache.

```
service start httpd
```

For more information on these options, see:

- `http://httpd.apache.org/docs/2.0/mod/mod_ssl.html`
- `http://httpd.apache.org/docs/2.0/mod/mod_headers.html`

## Verifying ICAT and MCAText through Apache

Again, using a web browser, test the ICAT and MCAText services via Apache.

- `https://localhost/icat/ws` should show two services: `icatService` and `srbNotifySOAP`
- `https://localhost/mcatext/ws` should show three services: `srbSyncSOAP`, `srbRegisterSOAP` and `srbAuthzSOAP`

### Troubleshooting

Check for SOAP communications between ICAT and MCAText recorded in the `catalina.out` log file.

If you encounter difficulties configuring Apache SSL, you can configure ICAT and MCAText to allow non-authenticated connections as follows:

1. In the deployed `icat.xml`, modify the `contextConfigLocation` parameter to read as follows:

   ```
   <Parameter name="contextConfigLocation" value="WEB-INF/beans-nosecurity.xml"
   override="false"/>
   ```
2. Make the same change to the deployed `mcatext.xml`.

You can now connect to ICAT and MCAText using HTTP on port 8080 or using HTTPS on port 443.

## Verifying GSI

To test that GSI authentication is working, use the ARCHER tool Hermes. Set it up to use GSI authentication as described in the Hermes user manual.

### Troubleshooting

Watching the `$CATALINA_HOME/logs/catalina.out` file, make a request from a GSI enabled client like Hermes, or the Python command line tools. Make sure the address starts with `https`.

You should see text similar to the following:

```
Headers: {Max-Forwards=[10], content-length=[517], accept-encoding=[identity],
host=[icatserver:443],
SOAPAction=["http://archer.edu.au/services/iCATService/getInvestigationById"],
content-type=[text/xml; charset=utf-8],
SSL_CLIENT_S_DN=[/C=AU/O=Grid/OU=Dev/CN=username]}
```

Look for the `SSL_CLIENT_S_DN` reporting the true DN of the connecting user.

If this is the case, ICAT has been correctly set up in Apache.

If not, Apache is not requesting peer verification, or bringing the SSL variables into scope in its configuration file, or is not setting the HTTP headers. See the Apache section above.

# Maintenance

## Stopping and starting

To stop Tomcat:

```
$TOMCAT_HOME/bin/shutdown.sh
```

To start Tomcat:

```
$TOMCAT_HOME/bin/startup.sh
```

To remove just one of the applications, stop Tomcat, then delete the context file and corresponding `webapps` directory from Tomcat:

```
rm -rf $TOMCAT_HOME/webapps/icat
rm $TOMCAT_HOME/conf/Catalina/localhost/icat.xml
```

To stop Apache:

```
service httpd stop
```

To start Apache:

```
service httpd start
```

## Logging

The Tomcat log files are found in `$TOMCAT_HOME/logs/Catalina.out` .

Apache's log files are in `/etc/httpd/logs` .

## Configuring

To reconfigure ICAT or MCAText, either:

1. Repeat the steps to generate the context files, and redeploy them; or
2. Directly modify the deployed context files. Some settings in these files are not documented.