ARCHER 1.0

Common Instrument Middleware Architecture & DIMSIM

System Administrator's Guide

- Installation
- Configuration
- Testing

About CIMA and DIMSIM

Overview

CIMA, the Common Instrument Middleware Architecture, is a plugin-based architecture to facilitate the connection of different kinds of scientific instruments. “Producer” plugins retrieve data parcels which are served over a network to “consumer” plugins.

DIMSIM is a pair of plugins for CIMA, designed to work with the Rigaku crystallography spectrometer:

- the DIMSIM “producer” reads files in CrystalClear 1.3[1] format from a directory, sending them across a network
- the DIMSIM “consumer” receives these parcels and writes them to an SRB.

Use of CIMA with other instruments would require development of a specific plugin. See the “Developing for CIMA” section.

A test harness is included with DIMSIM to facilitate testing and developing. It produces mock CrystalClear files which DIMSIM detects and processes.

Note:
This release of DIMSIM has not been tested and has not been used in a production environment. It has certain limitations, particularly in security and authentication, due to being developed with Version 1.3 of CrystalClear, rather than Version 1.4.

Network requirements

Before installing DIMSIM, you should install Archer Data Services Infrastructure Layer. This creates the SRB that the DIMSIM producer writes to.

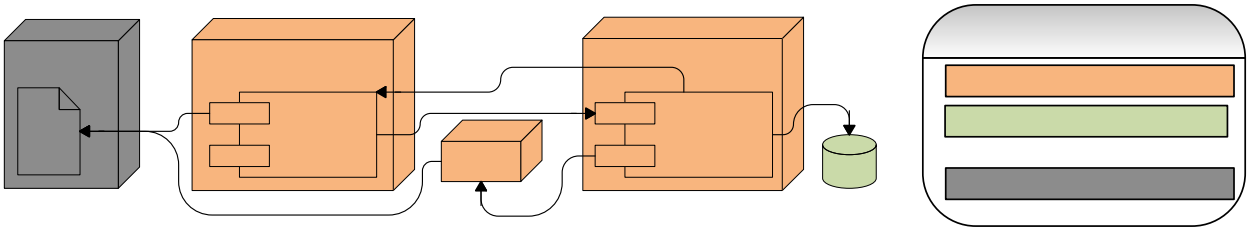
The **producer** machine must have:

- inbound access from the consumer machine on port 80 and 8080 (configurable).
- access to a directory where CrystalClear writes output files to (generally a Samba share).

The **consumer** machine must have:

- inbound network access from the producer machine on port 8081 (configurable)
- outbound network access to an SRB (port 5544 by default)

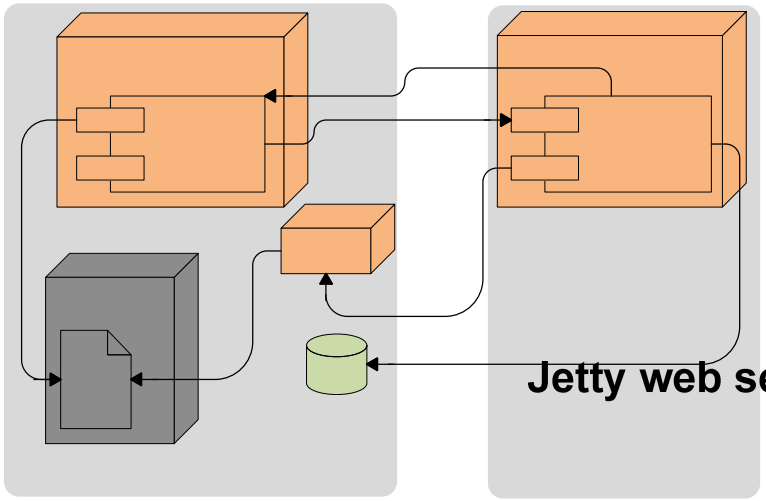
Both machines must have outbound access to the internet at installation time and the first time DIMSIM is run.



DIMSIM connectivity diagram

[1] http://www.rigaku.com/software/crystalclear.html

In the ARCHER development environment, the two machines were configured as follows.



ARCHER development environment configuration

This document assumes this kind of configuration.

Installing CIMA/DIMSIM – producer and consumer

The first few installation steps are identical, whether for installing the producer or the consumer. Then, the configuration depends on whether the machine is being set up as producer or consumer.

The following steps must be performed for **both** the consumer and producer installation.

Prerequisites

Install JDK version 1.5 or later, and Subversion, if not already installed:

```
yum install jdk subversion
```

Installing Maven 2

Maven 2 is required to both build and run DIMSIM. When launched, it retrieves other components from the internet to run CIMA as necessary.

Download it from http://maven.apache.org/download.html . Follow the steps on this page. In particular, ensure that the following environment variables are set, preferably from a login script such as `/etc/profile`.

Environment variable	Meaning
M2_HOME	The directory where Maven is installed.
M2	$M2_HOME/bin
JAVA_HOME	The directory where the JDK is installed.

```
cd
wget http://apache.mirror.aussiehq.net.au/maven/binaries/apache-maven-2.0.9-
bin.tar.gz
mkdir /usr/local/maven
cd /usr/local/maven
tar -xzf ~/apache-maven-2.0.9-bin.tar.gz

cat <<-EOF > /etc/profile.d/maven.sh
export M2_HOME=/usr/local/maven/apache-maven-2.0.9
export M2=\$M2_HOME/bin
export PATH=\$M2:\$PATH
EOF
```

The `PATH` must include `$M2` and `$JAVA_HOME/bin`:

```
cat <<-EOF > /etc/profile.d/java.sh
export JAVA_HOME=/usr/java/jdk1.5.0_16
export PATH=\$JAVA_HOME/bin:\$PATH
EOF
```

Installing Jetty

Jetty is a lightweight web and application server. It provides the transport used by both consumers and producers.

Create a user (not superuser) called `dimsim`. Change to this user.

```
adduser dimsim --password xxxxx
su - dimsim
```

Download the most recent stable Jetty 6 release from http://dist.codehaus.org/jetty/.

```
cd ~dimsim
wget http://dist.codehaus.org/jetty/jetty-6.1.11/jetty-6.1.11.zip
```

Unzip it to `/usr/local/jetty`. More information is available at `http://docs.codehaus.org/display/JETTY/Installing+Jetty-6.1.x` . The unzipped files should be owned by and run as dimsim.

```
sudo mkdir /usr/local/jetty
chown dimsim /usr/local/jetty
cd /usr/local jetty
unzip ~/jetty-6.1.11.zip
```

Downloading DIMSIM/CIMA source

Download and unzip the source code from http://www.archer.edu.au/downloads

```
cd
wget http://www.archer.edu.au/downloads/DIMSIM_1.0.tar.gz
tar –xzf DIMSIM_1.0.tar.tgz
```

Note: this tarball contains symlinked files. Hence, downloading on another platform, and transferring the files to the target machine will likely break these links.

Configuring DIMSIM as producer

Overview

The machine running the DIMSIM **producer** uses Apache HTTP Server to serve image files across the network to the consumer.

Installing Apache

Install Apache:

```
su
yum install httpd
```

Optional
You can configure Apache to listen on different port, by editing the line which reads "`Listen 80`" in `/etc/httpd/conf/httpd.conf`

Create a file called `/etc/httpd/conf.d/cima.conf` as follows:

```
cat <<-EOF > /etc/httpd/conf.d/cima.conf
Alias /xtal-files /tmp/pollertest

<Directory /tmp/pollertest>
        # for sharing files from a network mount
        EnableSendfile off

        Options Indexes
        AllowOverride None
</Directory>
EOF
```

This configuration shares the `/tmp/pollertest` directory as `/xtal-files`. This is required for the testing environment, but should be modified for any production environment.

Start the Apache server.

```
service httpd start
```

Verify that Apache is configured correctly as follows:

Command	**Purpose**
`mkdir /tmp/pollertest`	Create the test directory
`echo Testing Apache > /tmp/pollertest/test.txt`	Create a test file
`wget http://localhost/xtal-files/test.txt`	Retrieve the test file via Apache
`cat test.txt`	Verify its contents.
`rm /tmp/pollertest/test.txt`	Clean up.

Configuring producerPlugins.xml

The `producerPlugins.xml` file needs to be configured for the external address of the Apache server. This address is passed to consumers in order for them to download image files.

Modifying producerPlugins.xml

1. Modify the line which begins "`<property name="urlPrefix"…`" to match the external name of your Apache server. For example:

   ```
   <property name="urlPrefix" value="http://producer.uni.edu.au/xtal-files/"/>
   ```

   This name is used by the consumer, so do not use "localhost" unless the consumer is on the same machine.

2. Modify the line which begins "`<property name="urlprefixReplace"…`" to match the network share being served. When using the test harness, set it as follows:

   ```
   <property name="urlPrefixReplace" value="/tmp/pollertest/"/>
   ```

3. Modify the <constructor-arg> that immediately precedes the "pathReplacements" to point to the location of your CrystalClear scripts. For the test harness, modify it as follows:

   ```
   <constructor-arg type="java.lang.String"
   value="/tmp/pollertest/SessionScript.scp"/>
   <property name="pathReplacements">
   ```

4. When using the test harness, uncomment this section:

   ```
   <bean
   class="au.edu.archer.cima.plugin.xtal.SCPDirectorySinglePollerParcelProducer$Replac
   ement">
   <constructor-arg value="C:/Program Files/Rigaku
   MSC/CrystalClear/Data/rossjohn/SaBPL_inhibitors/050607"/>
   <constructor-arg value="/tmp/pollertest"/>
   ```

Configuring run_producer.sh

By default, the producer will run Jetty on port 8080. If this port is unavailable or undesirable, modify the file `run_producer.sh` (actually a symlink to `run_util.sh`) as follows.

Optional

`*producer*)`
`  export D_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/lib"`
`Cp producerPlugins.xml target/classes/plugins.xml`
`  MVN_OPTS="$MVN_OPTS -Djetty.port=8280"`
`;;`
	Add this line with a port as appropriate

If the default port is acceptable, you do not need to edit the file.

Configuring DIMSIM as consumer

Overview

The DIMSIM consumer subscribes to the producer, then awaits parcels. It then connects to the producer's Apache server to download the data, which it then saves to the SRB.

Preparing the SRB

The consumer must have access to the SRB. In many cases, it should be run on the same machine. You should create a dedicated SRB user for DIMSIM, probably called dimsim.

Option 1:
If you are using LDAP for SRB user authentication, create an LDAP user. This must be done on a machine with LDAP client tools installed.

For example:

```
smbldap-useradd -P dimsim
```

Option 2:
Alternatively, simply create a user directly in the SRB. Download MCATadmin (`http://www.sdsc.edu/srb/tarfiles/mcatAdmin.jar`) and use it to create a new user called dimsim.

Configuring consumerPlugins.xml

The `consumerPlugins.xml` controls how the consumer connects to the SRB.

Modify all the following properties with values appropriate for your environment.

Property	Meaning
`<property name="srbHost" value="srb.uni.edu.au"/>`	The host name of the SRB.
`<property name="srbUsername" value="dimsim"/>`	The login used to access the SRB.
`<property name="srbPassword" value="dimsim"/>`	Its password.
`<property name="srbMDASDomain" value="srb.uni.edu.au"/>`	The SRB domain.
`<property name="srbMDASZone" value="srb.uni.edu.au "/>`	The MCAT zone.
`<property name="srbDefaultResource" value="demoResc"/>`	The default resource, frequently "demoResc".
`<property name="usingX509" value="false"/>`	Whether X509 certificates are used. False in this case.

If you are not using the Labjack plugin, delete or comment out the `<ref bean="environmentSubscribe"/>` as follows:

```
<!-- <ref bean="environmentSubscribe"/> -->
```

The Labjack plugin is not used by the test harness.

Inside the "remoteEndpoint" bean, edit the innermost <xml-fragment> tag to contain a single <p:url> tag. It should point to /ws/cima on the Jetty server running on the producer machine. For example:

```
<xml-fragment type="SOAP"
xmlns:p="http://cima.instrumentmiddleware.org/parcel">
<p:url>http://producer:8080/ws/cima</p:url>
</xml-fragment>
```

Configuring run_consumer.sh

By default, the consumer Jetty server will start on port 8081. If this port is unavailable or undesirable, edit run_consumer.sh and modify this line:

```
MVN_OPTS="$MVN_OPTS -Djetty.port=8081"
```

Configuring consumer-cima-webservice.properties

If the consumer and producer are not running on the same machine, edit consumer-cima-webservice.properties to refer to the producer machine. For example:

```
cima.soap.localEndpointUrl=http://producer:8081/ws/cima
```

Preparing the experiment test harness

For testing purposes, a test harness is included that mimics the behaviour of the Rigaku crystallography spectrometer. It copies image files to a directory which DIMSIM has been configured to recognise as a Windows Samba share.

The test harness is set up on the **producer** machine.

```
mkdir --parents /tmp/pollertest/originals
mkdir /tmp/pollertest/Images
```

Copy a sample `.osc` file as `/tmp/pollertest/image.osc`.

```
cp ~/image.osc /tmp/pollertest/image.osc
```

A sample .osc file can be obtained from http://www.archer.edu.au/downloads.

Running DIMSIM

To run the DIMSIM test harness, the steps are as follows:

1. Run the producer.
2. Run the consumer.
3. Run the test harness script.

Running the DIMSIM producer

On the **producer** machine, run the `run_producer.sh` script as the dimsim user.

The first time you run this script, use the `-u` option. This tells maven to download other components to complete the installation. This takes several minutes.

On subsequent runs, you can remove the `-u` option to improve start-up time.

```
su - dimsim
cd CIMA
mkdir --parents target/classes
./run_producer.sh -u
```

Verifying

The following lines should appear in the output.

```
[TRACE] (ListPluginManager.java:59) now have 3 plugins
…
[TRACE] (CIMAUtil.java:170) loading 2 transports
…
[TRACE] (ListTransportManager.java:50) now have 2 providers
…
```

The final line should be:

```
[INFO] Started Jetty Server
```

Running the DIMSIM consumer

On the producer machine, run the `run_consumer.sh` script as the dimsim user.

This must also be run with the –u option the first time.

```
su dimsim
./run_consumer.sh -u
```

Note: The producer must always be started before the consumer, due to the subscription service. If the producer is restarted, the consumer must also be restarted.

Verifying:

On the consumer machine, a burst of output should be generated, terminating in `[INFO] Started Jetty Server`

On the producer machine, a burst of output should be generated, including: `[TRACE] (SCPDirectorySinglePollerParcelProducer.java:132) processing subscribe parcel`

The producer will then sporadically produce output as it polls for image files.

Troubleshooting:
If errors result, shut down the producer, consumer and test harness. Double check all configuration files. Clean all files from `/tmp/pollertest` and `/tmp/pollertest/Images` on the producer machine.

Running the DIMSIM test harness

Once the consumer and producer are running, run the experiment test harness script, `run_experiment.sh`, on the **producer machine**:

```
./run_experiment.sh -c 3 -d 5
```

This will cause 3 image files to be copied, and hence for DIMSIM to transmit them across the network eventually to the SRB. They will be sent at 5 second intervals.

Run `run_experiment.sh -h` for help on the parameters.

Verifying:
This output should be produced in the experiment window:

```
rm: cannot remove `/tmp/pollertest/Images/*.osc': No such file or directory
cp: cannot stat `/tmp/pollertest/SessionScript.scp.orig': No such file or
directory
created 1.osc
created 2.osc
created 3.osc
1) restart experiment
2) set image count
3) set image delay
#?
```

Troubleshooting
On the consumer machine, you may also need to replace the Jargon .jar file with a later version. In particular, `.m2/repository/SDSC/jargon/2.0.0beta/jargon-2.0.0beta.jar` should be replaced with `jargon-2.0.1beta.jar`. This file should be copied over the old one.

Developing for CIMA

CIMA is an extensible framework which allows other plugins to be developed for it. Documentation for the CIMA API is available at http://www.archer.edu.au/downloads.