Science & Technology Facilities Council

Getting started – how to set up the common data tables

NOBUGS 2012 ICAT Workshop

27th September 2012

Kevin Phipps

Scientific Computing Department, RAL

What needs to be in place?

- ICAT (4.2) installed and running on an application server such as Glassfish

- An authentication plugin
  - 2 standard versions available: DB and LDAP
  - Download from http://www.icatproject.org/mvn/site/

Using the database authentication plugin

- When the authentication ear file is deployed a database table PASSWD is created
- 2 columns USERNAME and ENCODEDPASSWORD
- Add a username and a password – the password must be in plain text and not encoded as the column name suggests. This will change in a later version.

Connecting to ICAT

- ICAT runs as a web service and is described by its WSDL document such as the one at: http://www.icatproject.org:8080/ICATService/ICAT?wsdl
- To connect using Java use the client library at: http://www.icatproject.org/mvn/site/icat/4.2.0/icat.client/installation.html
- Alternatively generate your own client using a tool such as NetBeans or connect in Python via the suds library by pointing it at the WSDL URL

Making the initial connection

```
URL hostUrl = new URL("http://localhost:8080");
URL icatUrl = new URL(hostUrl,"/ICATService/ICAT?wsdl");
QName qName = new QName("http://icatproject.org", "ICATService");
ICATService service = new ICATService(icatUrl, qName);
ICAT icat = service.getICATPort();
```

- Keep hold of the reference to “icat”. You will need it for every call you want to make to the server.

Setting up login credentials

```
Credentials credentials = new Credentials();
List<Entry> entries = credentials.getEntry();
Entry e;
e = new Entry();
e.setKey("username");
e.setValue("admin");
entries.add(e);
e = new Entry();
e.setKey("password");
e.setValue("secret");
entries.add(e);
```

Login

- Pass the credentials to the login method specifying the authentication mechanism to use

```
String sessionId = icat.login("db", credentials);
```

- Hold on to the sessionId. It is required when making subsequent calls to ICAT.

Setting up basic authorisation

- Add the admin user you have set up to ICAT

```
User adminUser = new User();
adminUser.setName("admin");
adminUser.setId(icat.create(sessionId, adminUser));
```

- Set up an ICAT Group

```
Group rootGroup = new Group();
rootGroup.setName("root");
rootGroup.setId(icat.create(sessionId, rootGroup));
```

- Add the admin user to that group

```
UserGroup userGroup = new UserGroup();
userGroup.setUser(adminUser);
userGroup.setGroup(rootGroup);
icat.create(sessionId, userGroup);
```

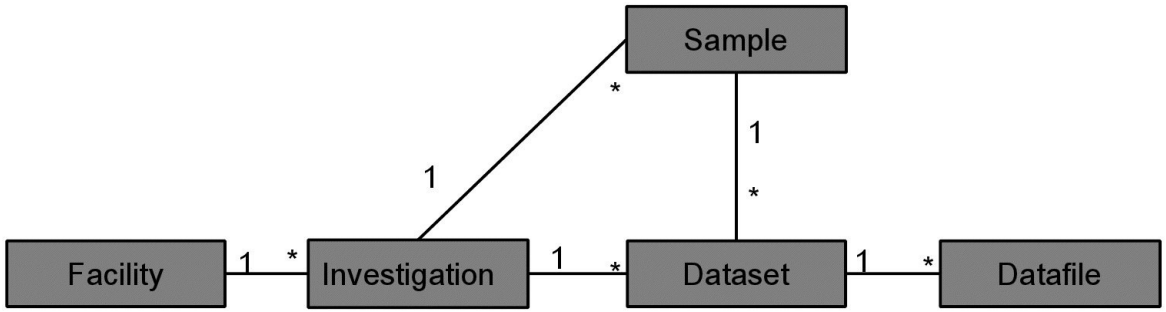
Assigning privileges

- Create a rule – the example below gives our user “admin” Create, Read, Update and Delete privileges on Facility objects

```
Rule rule = new Rule();
rule.setGroup(rootGroup);
rule.setWhat(“Facility”);
rule.setCrudFlags(“CRUD”);
icat.create(sessionId, rule);
```

- Add further rules for each type of object to be created

Schema - The main components

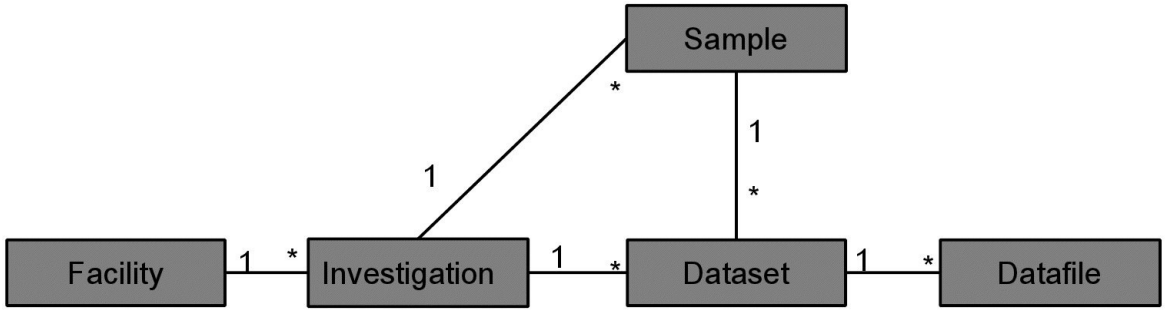


- Facility – an experimental facility
  ICAT must contain one before you can add any data.

- Investigation – an investigation or experiment
  This is typically defined by a group of users visiting an experimental facility for a defined period of time during which they create data they want to save.

Science & Technology Facilities Council

Main components (cont.)



- Dataset – a collection of data files and parameters

  Typically contain a number of DatasetParameters (to record settings, times, temperatures etc) and a list of associated Datafiles.

- Datafile – a file and associated parameters

  Examples of these are configuration files, image files, binary data files. Each Datafile can have a list of DatafileParameters associated with it.

Creating a Facility in ICAT

```
Facility facility = new Facility();
facility.setName("ISIS");                                        // required
facility.setFullName("ISIS Pulsed Neutron and Muon Source");     // optional
facility.setId(icat.create(sessionId, facility));
```

Science & Technology Facilities Council

Creating an Investigation

- An InvestigationType must be created first

```
InvestigationType invType = new InvestigationType();
invType.setFacility(facility);
invType.setName("calibration");
invType.setDescription("A calibration experiment");
invType.setId(icat.create(sessionId, invType));
```

- Then use it to create the Investigation

```
Investigation inv = new Investigation();
inv.setFacility(facility);
inv.setName("Investigation 1");
inv.setTitle("A more grand title for Investigation 1");
inv.setType(invType);
inv.setId(icat.create(sessionId, inv));
```

Creating a Dataset

- A DatasetType must be created first

```
DatasetType rawDsType = new DatasetType();
rawDsType.setFacility(facility);
rawDsType.setName("experiment_raw");
rawDsType.setDescription("RAW data collected during an experiment");
rawDsType.setId(icat.create(sessionId, rawDsType));
```

- Then use it to create Datasets of that type (Covered in more detail later in the Workshop)

Creating a Datafile

- A DatafileFormat must be created first

```
DatafileFormat nexusFormat = new DatafileFormat();
nexusFormat.setFacility(facility);                    // required
nexusFormat.setName("nexus");                         // required
nexusFormat.setVersion("4.3.0");                      // required (I think!)
nexusFormat.setType("HDF5");
nexusFormat.setDescription("ISIS Muon format");
nexusFormat.setId(this.icat.create(this.sessionId, nexusFormat));
```

- Then use it to create Datafiles of that type
  (Covered in more detail later in the Workshop)

Adding parameters to objects

- A ParameterType must be created first

```
ParameterType protonChargeType = new ParameterType();
protonChargeType.setFacility(facility);                    // required
protonChargeType.setName("total_proton_charge");           // required
protonChargeType.setUnits("uAh");                          // required
protonChargeType.setValueType(ParameterValueType.NUMERIC); // required
protonChargeType.setApplicableToDatafile(true);
protonChargeType.setApplicableToDataset(true);
protonChargeType.setApplicableToInvestigation(true);
protonChargeType.setApplicableToSample(true);
protonChargeType.setDescription("The total proton charge on target during the
      collection period");
protonChargeType.setUnitsFullName("micro amp hours");
protonChargeType.setEnforced(true);
protonChargeType.setMinimumNumericValue(new Double(0));
protonChargeType.setVerified(true);
protonChargeType.setId(icat.create(sessionId, protonChargeType));
```

Basic setup complete

- ICAT contains a Facility, an Investigation, some DatasetTypes, ParameterTypes and DatafileFormats
- ICAT is now ready to ingest Datasets and Datafiles
- Further configuration required for a more extensive setup using FacilityCycles, Instruments, Applications etc but they are all done in the same way

And finally...

- Any questions?
- Discussion about possibility of agreeing a basic set of Types and Formats?

- And thanks - to Tom Griffin for providing the ISIS related examples used in this presentation

Dataset_Type

NAME	DESCRIPTION
analyzed	Analyzed data
experiment_cal	RAW data collected at the facility during a calibration run.
experiment_eng	RAW data collected at the facility during a engineering test.
experiment_raw	RAW data collected at the facility during an experiment.
reduced	Reduced Data
simulation	Simulation data
special_cal	"Calibration data not acquired through the normal Data acquisition system.

Investigation_Type

NAME	DESCRIPTION
calibration	A set of Calibration
commercial_experiment	A scientific experiment performed by a commercial company
engineering	"Calibration, first light data, alignment, ?"
experiment	A scientific experiment.
simulation	

Datafile_Format

NAME	VERSION	FORMAT_TYPE	DESCRIPTION
nexus	3.0.0	HDF5	Neutron and X-Ray data format.
isis neutron raw v8	8	binary	
isis muon raw v4	4	binary	
nexus	1	binary	
nexus	2.1.0	HDF4	ISIS Muon format
isis neutron raw	2	binary	
nexus	4.3.0	HDF4	ISIS Muon fomat
nexus	4.1.0	HDF4	ISIS Muon format