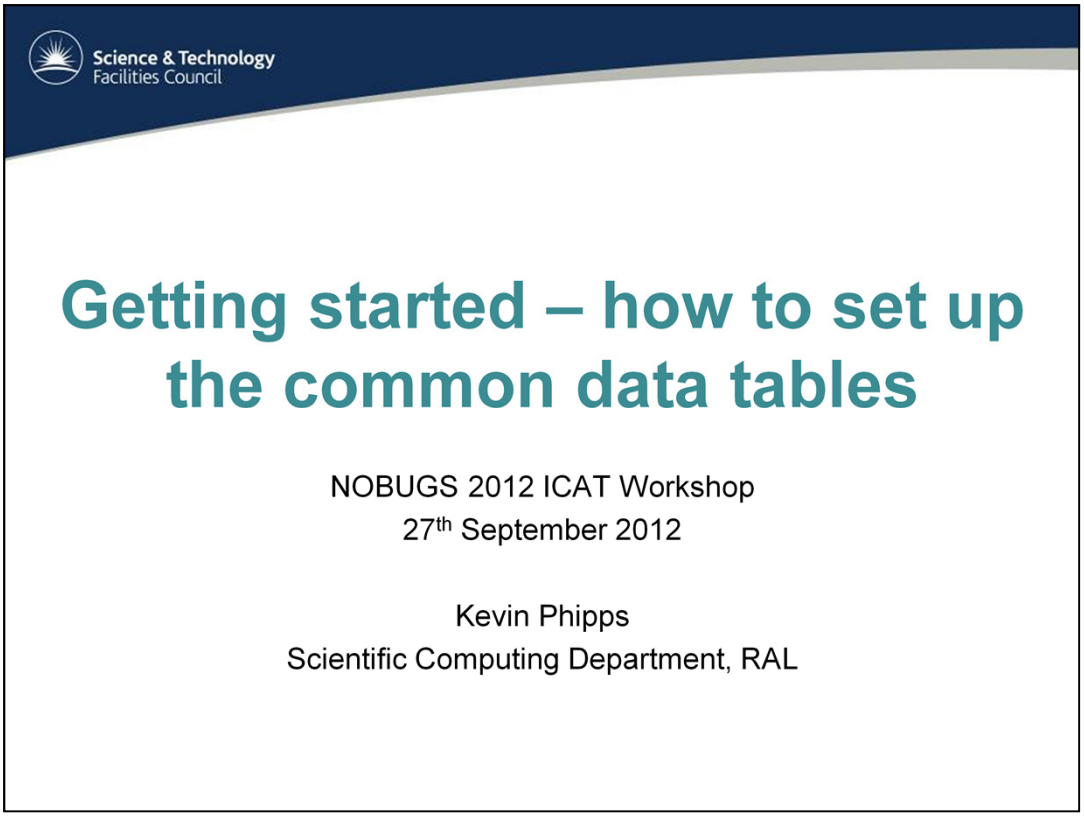


This presentation will explain the basics of how to get started with a newly installed and empty ICAT.

There is a certain amount of setting up which is mostly specific to your facility, your experiments and the type of data you are storing, that needs to be set up in ICAT first.

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/

Science & Technology Facilities Council

So what do you need to have in place before you start.

ICAT needs to be installed and running on an application server such as Glassfish.

ICAT 4.2 is assumed for the examples given in this presentation.

You also need an Authentication plugin.

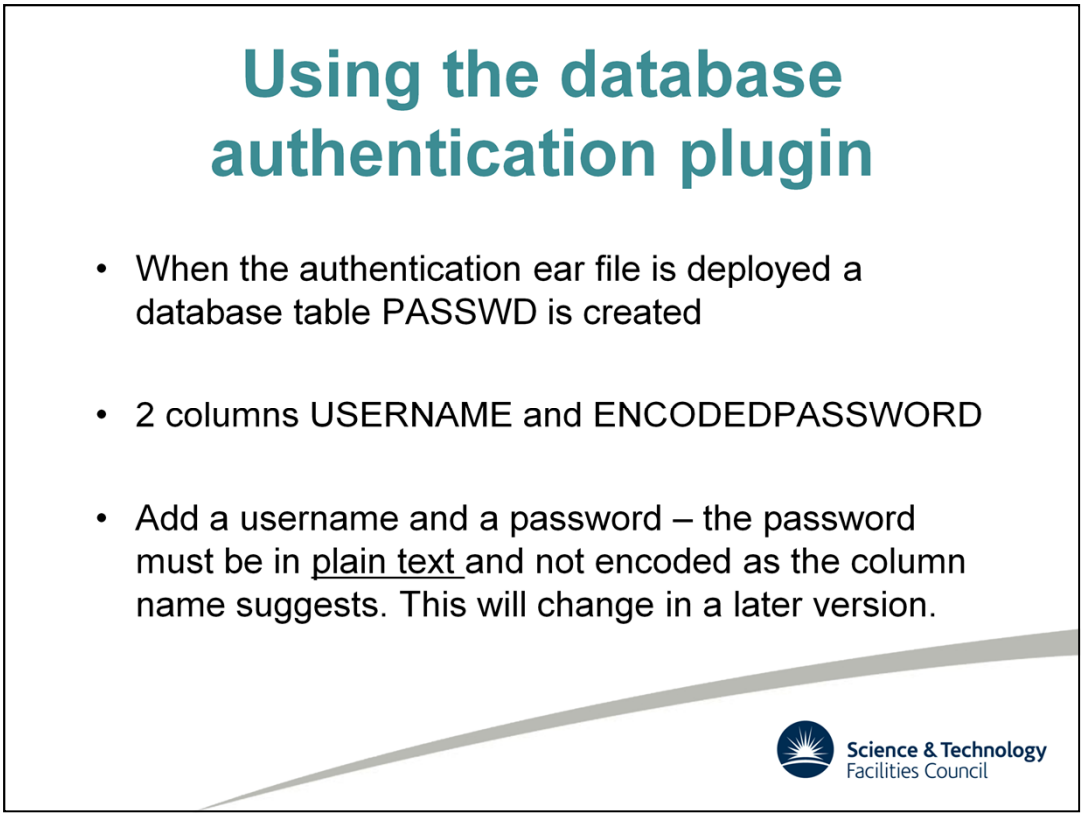
This is configured in a new way in ICAT 4.2.

It is deployed as an EJB in your application server and located via JNDI

There is a DB or 'database' version which runs on a relational database and an LDAP version for LDAP databases.

These plugins are available to download from the URL shown.

They will be covered in more detail later in the workshop.



To keep things simple I will just be talking about how to use the database authentication plugin.

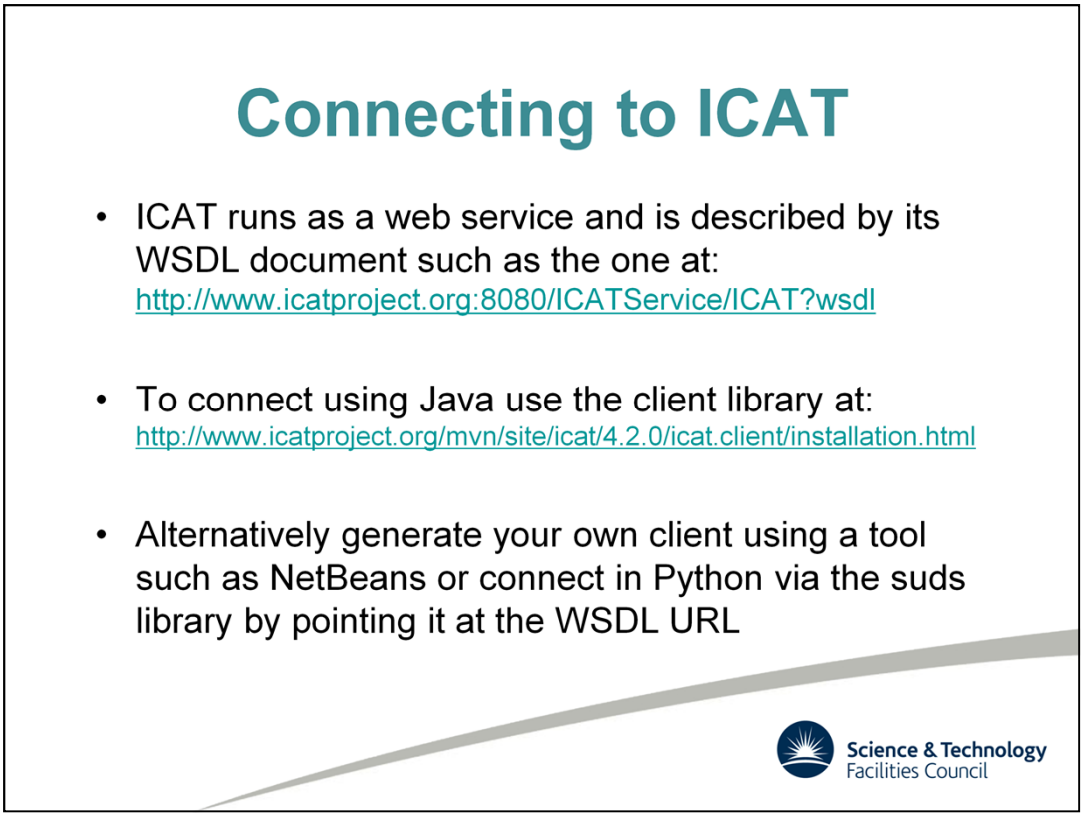
For this plugin, when the authentication ear file is deployed in your application server a database table PASSWD is created

2 columns called USERNAME and ENCODEDPASSWORD are created in this table.

You need to add a username and password using the database tool of your choice.

The password must be in plain text and not encoded as the column name suggests

This will change in a later version.



So how do we get connected to ICAT.

ICAT runs as a web service and is described by its Web Service Definition Language document such as the one at the URL shown.

The WSDL document is an XML file describing all the methods that can be called on the web service and all the datatypes used by those methods.

To connect using Java use the client library at the second URL shown.

The code snippets in this presentation use the standard Java client library.

Alternatively you can generate your own client using a tool such as NetBeans or connect in Python via the suds library by simply 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.

Here is our first snippet of code.

All it’s doing is specifying the URL where the web service is running and passing it to the client library.

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);
```

Science & Technology Facilities Council

With the new authentication plugins mechanism a Credentials object needs to be set up.

This example looks slightly over-complicated for a simple username/password login but this method does support many other kinds of login as well.

In this case all that is being done is adding two key-value pairs to a list – a username of ‘admin’ and a password of ‘secret’.

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.

Science & Technology Facilities Council

To do the actual login, simply pass the credentials to the login method specifying the authentication mechanism to use as shown in this code snippet

You also need to 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);
```

Science & Technology Facilities Council

The first thing you need to do having logged in and got yourself a session ID is to set up some users and groups.

In version 4.0 and 4.1 of ICAT there had to be a super user called root in the root group.

In version 4.2 the root group still exists but you can add usernames of your choice to it.

So in the example here, we create a User object, set the name “admin” on it and then call the icat create method.

Then we create a Group object, set the group name of “root” on it and call the icat create method.

The return value from the create method is the ID of the object created in ICAT so setting the ID of the object in the client allows it to be used without having to make a further call to retrieve it from ICAT.

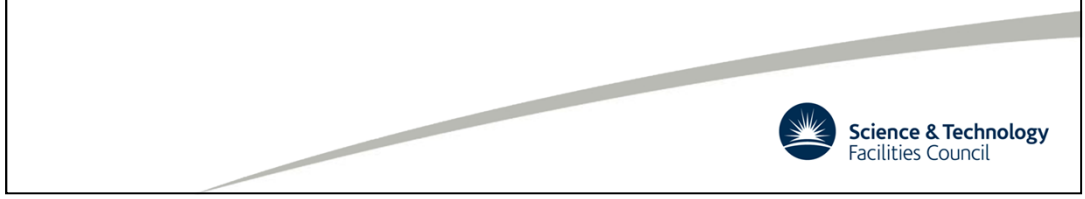
Finally, we combine the User and the Group object into a UserGroup, and the user admin has been added to the root group.

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



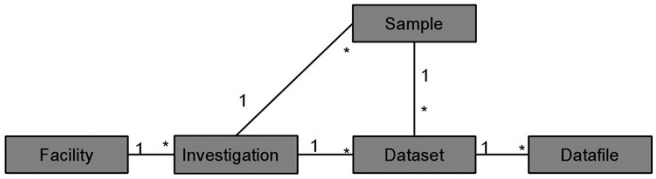
Science & Technology Facilities Council

To be able to do anything more with ICAT the next thing you need is some authorisation rules.

The example here shows how to set up a rule allowing users in the root group all privileges on Facility objects

You then need to 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

Coming back to the schema, here is a reminder of the main schema components.

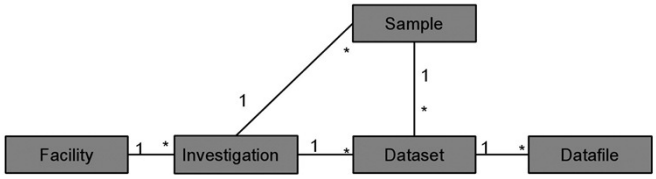
Facility is the root entity and represents an experimental facility.

There must be at least one Facility set up in ICAT before you can add any data.
As of ICAT 4.2, it is now possible to have multiple facilities set up in a single ICAT.

Investigation represents 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.

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.

Science & Technology Facilities Council

A Dataset represents a collection of data files and parameters.

It is up to each facility to define what they want a Dataset to contain.

It may be data collected over a certain period of time or the data collected when a particular event occurred, for example, a sample being scanned.

A Datafile represents a file and its associated parameters

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

Sample will not be covered in this presentation because I do not have any experience of using it myself.

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

Lets see some examples of creating objects in ICAT.

There are always mandatory fields that have to be set because they correspond to relationships in the database, and optional fields which you may find useful for other purposes.

In this example of setting up a Facility the only required field is the Facility name, which is typically the short name or acronym by which the facility is known.

Setting the full name would allow tools such as TopCat to provide a more detailed description if they wish to.

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));
```

Science & Technology Facilities Council

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)

Science & Technology Facilities Council

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)

Science & Technology Facilities Council

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));
```

Science & Technology Facilities Council

Before you can add parameters to objects, some ParameterTypes need to be set up.

This is a slightly more detailed example but as you can see, only the first 4 fields actually need to be set – the rest are optional.

A ParameterValueType needs to be set. The options are NUMERIC, STRING or DATE_AND_TIME.

You also need to specify which of the entities the ParameterType can be applied to.

The options are Samples, Investigations, Datasets and Datafiles.

In this example, the parameter can be used with all of these entities.

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

Science & Technology Facilities Council

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

Science & Technology Facilities Council

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

Science & Technology Facilities Council