SSO Client Tool

The SSO gateway hosts a central repository of all the user’s authentication and authorization information. This document describes how to use the SsoDbClient to access the SSO data. In this document, role is the information stored as a tuple of “mission:group:privilege”. Everything describe in this document is case-sensitive. The information must match exactly. The client is written in Java. All of the necessary libraries are bundled into the ssodb\_client.jar file. The only requirement is Java runtime.

## Usage

java -jar ssodb\_client.jar [-v] [-ops|-test] [-import=<data\_file\_name> | -lu[=<user>] | -lr[=<mission>] | -la[=<mission>] ]

-import: import the data file into the sso database");

Type header format is: \Type=data[:action]");

data may be 'user', 'role', or 'access'");

action may be 'add' or 'delete', default to 'add'");

-lu[=<user>]: list users. if user is provided, it will display user's info including his/her access.

-lr[=<mission>]: list roles. filter by mission if provided.

-la[=<mission>]: list access. filter by mission if provided.

-v: version

-b: brief or short format

DB connection is optional. if not given, it will prompt you.

-ops: connect to ops database

-test: connect to test database

### **-import**

Use this option to import data from an IPAC table file into the SSO system. Read the Input File Format section below for details on the file format.

### **Query options: -lu, -lr, and -la**

The query options allow you to retrieve data from the SSO system. The output format is in IPAC table. The same IPAC table can be use to import the data.

### **-lu**

Display users information. Use “=<user>” to only display information for that one user. In this mode, it will display the user’. <user> is the user’s login name. A user’s login name is his/her email address. This is true for all users except those “legacy” users with different login name than email.

### **-lr**

Display the roles information. Use “=<mission>” to limit the results to only that mission.

### **-la**

Display access privileges for all users. Use “=<mission>” to limit the results to only that mission.

### **DB connection options**

There are currently two instances of the SSO gateways running; one in the test environment, and the other in the operation environment. For convenience, the database connection information are compiled into this tool. To access the “test” version, use “-test”. To access the “operation” version, use “ops”. If you run this tool without this option, it will prompt you for the database connection information, like user id, password, and such. If you have these information in your environment, this tool will use that instead. The environment variables applicable are:

josso.db.driver

josso.db.url

josso.db.userId

josso.db.password

## Input File Format

The input file is an IPAC table with headers and columns. Headers provide crucial information on how to import the data. Read the headers section carefully, because it may lead to importing the wrong data.

In general, the tool will import the data if valid. When it cannot, it will log the error then continue. There is no rollback capability.

**Headers**

* \Type=type[:action] -- Type of data to import
  + this is a mandatory header. Without it, the tool will stop.
  + type can be one of ‘user’, ‘role’, or ‘access’.
  + optionally, you can append :<action> to the end of Type to modify its import behavior. Currently, action can be “add”, “update”, or “delete” with “add” being the default value.
* \AutoFill=[true|false] -- Automatically fill in the missing data
  + default to false
  + this only apply to ‘role’. When setting to true, it will automatically fill in the mission\_id and group\_id if not given.
* \<col\_name>.value=default\_value -- the default value for col\_name if one is not given
  + this is useful when adding a lot of information for a particular mission.

**User input file**

Email and password are mandatory. The rest are optional. If the user exists in the system, it will update the user’s information instead of adding it. When adding a new user, if password is not given, “password” will be used as the user’s password.

\Type=user

|login\_name|address|city|country|first\_name|institute|last\_name|phone\_number|postcode|password|

**Role input file**

This will add new role(s) into the system or remove existing ones. If AutoFill is not set, you must provide all of the required data to define a valid role. More information on the definition of a valid role is available in a different document.

If AutoFill is set to true, it will try to automatically fill in the missing mission\_id or group\_id. The algorithm used is the same for both mission\_id and group\_id. It will lookup the ID based on the name. If one exists, it will use that. If one does not exist, it will take the max value in the system and increment it by 1. A unique role is a combination of its mission\_name, group\_name, and privilege.

\Type=role

\AutoFill=true|false

|mission\_name|mission\_id|group\_name|group\_id|privilege|

**User’s access input file**

Use this to assigns a user to one or more roles. It will also remove access privilege for the given user, if action is ‘delete’. Use email to identify a user. Provide as much information as needed to define a unique role. If a unique role cannot be attains from the given information, the tool will log the error, and then continue.

\Type=access

|login\_name|mission\_name|mission\_id|group\_name|group\_id|privilege|

**Sample input files**

<roles.tbl>

\Type=role

\mission\_id.value=8

\mission\_name.value=SAMPLE

|group\_name |group\_id |

2012-group1 20121 added as <SAMPLE(8):2012-group1(20121):>

2012-group2 20122 added as <SAMPLE(8):2012-group2(20122):>

2012-group3 20123 added as <SAMPLE(8):2012-group3(20123):>

2012-group4 20124 added as <SAMPLE(8):2012-group4(20124):>

<roles-1.tbl>

\Type=role

\AutoFill=true

|mission\_name|mission\_id|group\_name |group\_id|

SAMPLE -1 2012-group8 added as <SAMPLE(8):2012-group8(20125):>

SAMPLE 2012-group9 -1 added as <SAMPLE(8):2012-group9(20126):>

SAMPLE 101 2012-group7 ERROR: mission\_id should be 8, not 101

<role\_mappings.tbl>

\Type=access

\mission\_id.value=8

\mission\_name.value=SAMPLE

|login\_name |group\_name |group\_id |

loi@ipac.caltech.edu 2012-group1 added to <SAMPLE(8):2012-group1(20121):>

loi@ipac.caltech.edu 20122 added to <SAMPLE(8):2012-group2(20122):>

loi@ipac.caltech.edu 2012-group4 20123 ERROR: cannot find a role with group

name/id combo

\Type=user

\Type=user

|login\_name |first\_name |last\_name|password|address|city|country|institute|phone\_number|postcode|

jd@acme.com john doe secret addr acme 555-2121

Add this user into the system. Will skip if this user already exists.

\Type=user:delete

\Type=user

|login\_name |first\_name |last\_name|password|address|city|country|institute|phone\_number|postcode|

jd@acme.com john doe secret addr acme 555-2121

Remove this user from the system.