

	TAP_SCHEMA Manager 1.0.1 User Guide	<table border="1"> <tr> <td>Document No.</td> <td>TSM-UG 1.0.1</td> </tr> <tr> <td>Issue/Rev. No.</td> <td>1.0.1</td> </tr> <tr> <td>Date</td> <td>Dec 1, 2016</td> </tr> <tr> <td>Page</td> <td>1 of 8</td> </tr> </table>	Document No.	TSM-UG 1.0.1	Issue/Rev. No.	1.0.1	Date	Dec 1, 2016	Page	1 of 8
Document No.	TSM-UG 1.0.1									
Issue/Rev. No.	1.0.1									
Date	Dec 1, 2016									
Page	1 of 8									

TAP_SCHEMA Manager 1.0.1 User Guide

December 14, 2016

Document No.: TSM-UG 1.0.1
Issue/Rev. No.: 1.0.1
Date: December 1, 2016
Authors: Sonia Zorba, Marco Molinaro
Approved by: Riccardo Smareglia

Contents

1	Introduction	2
2	Requirements	2
2.1	Java	2
2.2	Application server	2
2.3	RDMS	2
3	Installation	2
3.1	Create a proper SQL user	3
3.2	MySQL	3
3.3	PostgreSQL	3
4	Naming convention	4
5	Usage	4
5.1	Login	4
5.2	Credentials insertion	4
5.3	Schemata selection	5
5.4	Checking for inconsistencies	6
5.5	TAP_SCHEMA editing	6
5.5.1	UCD validation tool	7



TAP_SCHEMA Manager 1.0.1

User Guide

Document No.	TSM-UG 1.0.1
Issue/Rev. No.	1.0.1
Date	Dec 1, 2016
Page	2 of 8

1 Introduction

TAP_SCHEMA is a particular database schema defined in the Table Access Protocol (TAP) IVOA Standard. It is intended for containing metadata related to a TAP service, like schemata, tables and columns exposed by the service or datatype, indexing, description and other information about columns.

The TAP_SCHEMA Manager is a Java EE application that provides a graphical user interface that can be used to create or edit a TAP_SCHEMA also by people that don't have specific SQL skills.

TAP_SCHEMA Manager simplest set up consists in installing it on the same machine that is storing the database from which you want to create a TAP_SCHEMA and that will store the TAP_SCHEMA itself. However, TAP_SCHEMA Manager can act as a client for an arbitrary numbers of database servers. In that case you have to use the TAP_SCHEMA Manager with a SQL user having proper permissions (see [subsection 3.1](#)).

2 Requirements

2.1 Java

TAP_SCHEMA Manager needs Java 7 or higher.

2.2 Application server

TAP_SCHEMA Manager needs to be run inside a Java EE application server.

Currently TAP_SCHEMA Manager supports:

- GlassFish (4.1+);
- Tomcat (tested on version 8);

Note: we have done more testing on GlassFish.

Other application servers could be easily supported including proper libraries, so if you need to run it on a specific server asks developers.

2.3 RDMS

Currently TAP_SCHEMA supports:

- MySQL (tested on version 5.6);
- PostgreSQL (tested on version 9.3.14);

Note: we have done more testing on MySQL.

3 Installation

1. Unzip the TAP_SCHEMA Manager package:

```
unzip tsm-1.0.1-<container>.zip
```

2. Edit the `webapp.properties` file. Properties you can edit are:

- `ucd_service_url`: URL of the UCD web service (for validating UCDs, see [subsubsection 5.5.1](#)); if this URL is not reachable the application will work anyway, but without the UCD validation feature;



TAP_SCHEMA Manager 1.0.1

User Guide

Document No.	TSM-UG 1.0.1
Issue/Rev. No.	1.0.1
Date	Dec 1, 2016
Page	3 of 8

- `credentials_config_path`: path of a file that will be used by the TAP_SCHEMA Manager to store credentials inserted by users (this file doesn't need to exist but it has to be writable by GlassFish/Tomcat);
- `password`: password for accessing the TAP_SCHEMA Manager.

3. Execute the `update-war.sh` script.

4. Deploy the war package in the application server.

3.1 Create a proper SQL user

TAP_SCHEMA Manager has to login into a database server with an user that has to be able to:

- read (`SELECT`) from all schemata that you want to add to the TAP_SCHEMA;
- create a new schema (the TAP_SCHEMA schema), create new tables on that schema and insert or delete rows;

If you installed the TAP_SCHEMA Manager on the same machine hosting the database on which you want to work you can simply use the TAP_SCHEMA Manager with credentials that you usually use. Otherwise consider that you would need to create a proper user allowed to access the machine hosting the database from the machine hosting the TAP_SCHEMA Manager.

3.2 MySQL

A typical set up could be created with the following statements:

```
CREATE USER 'tsm'@'<tsm-host>' IDENTIFIED BY '<tsm-password>';
GRANT SELECT ON *.* TO 'tsm'@'<tsm-host>';
GRANT ALL PRIVILEGES ON `<tap_schema>`.* TO 'tsm'@'<tsm-host>';
```

If you are stuck on connection problems check also the content of MySQL configuration file (usually located at `/etc/mysql/my.cnf` or `/etc/my.cnf`), looking in particular for the following options:

- `bind-address`
- `skip-networking`

Important note if your TAP_SCHEMA references some views built using UDFs (User Defined Functions) the MySQL user used by the TAP_SCHEMA Manager needs to have grants allowing using those UDFs.

3.3 PostgreSQL

PostgreSQL roles configuration is quite complex so we suggest you accessing the database from the TAP_SCHEMA Manager using the standard `postgres` user.

You may have to configure the `pg_hba.conf` file (typical location for this files could be `/var/lib/pgsql/1/data/pg_hba.conf` or `/etc/postgresql/9.x/main/pg_hba.conf`), adding the allowed IP address with the `md5` method.

TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	all	192.168.0.5/32	md5

4 Naming convention

Unlike the standard, our TAP_SCHEMA Manager allows the TAP_SCHEMA to have an arbitrary name and to be located both on a different database server and a different machine from the source schema, although TAP-1.1 goes this same direction. This implies also the support for multiple TAP_SCHEMA schemata on the same database server.

To distinguish these 2 database servers we chosen the following naming convention:

- **Source database server:** the machine hosting the database referenced by the TAP_SCHEMA.
- **TAP_SCHEMA database server:** the machine hosting the database containing the TAP_SCHEMA schema.

In the simplest configurations these 2 machine are the same machine.

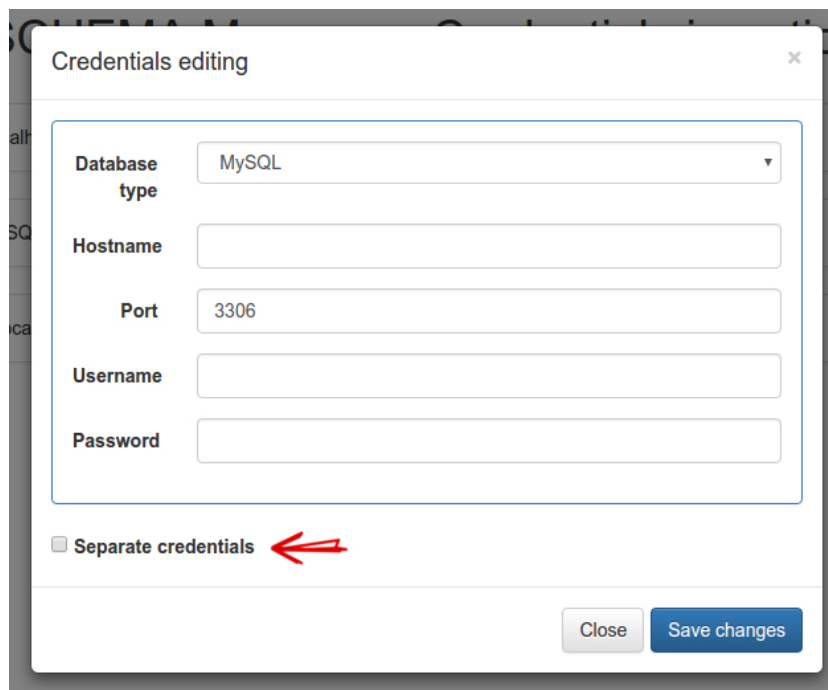
5 Usage

5.1 Login

On the first page the TAP_SCHEMA Manager asks you for the password you had configured in the installation step.

5.2 Credentials insertion

If you login successfully a page called “Credentials insertion” is shown to you. Here you can add an arbitrary number of credentials the TAP_SCHEMA Manager will use for connecting to databases.



Credentials editing
✕

Database type

Hostname

Port

Username

Password

Separate credentials ←←

Close
Save changes

Figure 1: Credentials editing modal

For adding a new credentials press the “Add new” button. A modal window will open and you can insert in it information that will be used for connecting to database servers. If the source database server is different from the TAP_SCHEMA database server you need to select the “Separate credentials” checkbox (see [Figure 1](#)).

When you press the “Save changes” button, credentials will be permanently stored in the file you have specified in the `credentials.config.path` setting.

Each saved credential will appear in a panel ([Figure 2](#)).

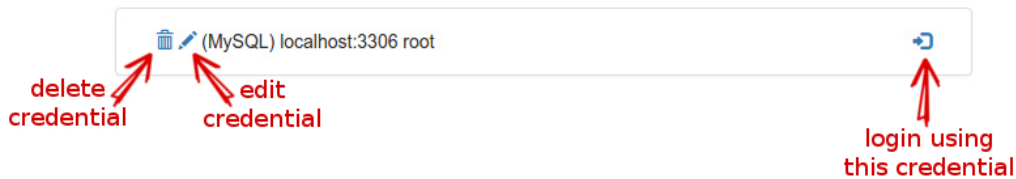


Figure 2: Use a saved credential

5.3 Schemata selection

When you press the “Login in” button from a saved credential panel you will reach a page called “Schemata selection”.

Here you can choose between edit an existing TAP_SCHEMA or create a new one. You can switch between this 2 actions using a radio button ([Figure 3](#)).

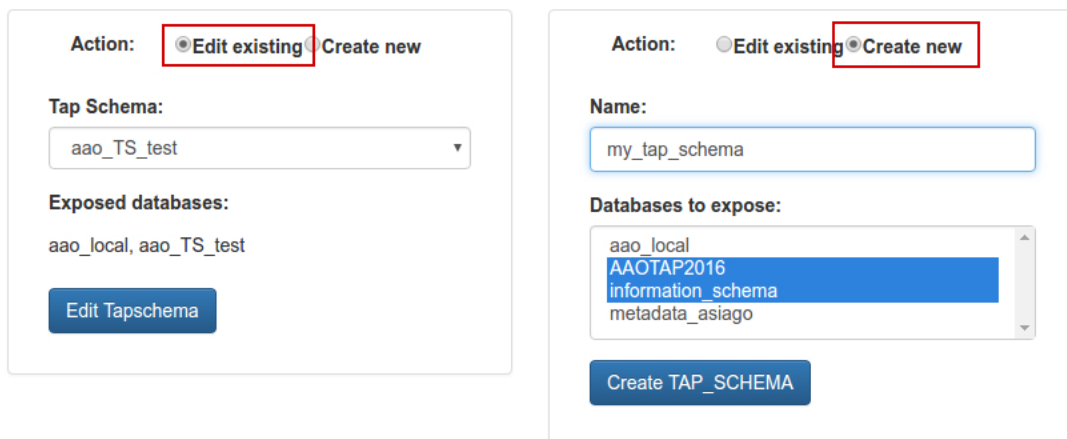


Figure 3: Edit an existing TAP_SCHEMA or create a new one

If you are choosing to create a new TAP_SCHEMA you can specify one or more schemata to include into the TAP_SCHEMA. Schemata selected in this phase will be added to the TAP_SCHEMA on the initial setup, but you can add more schemata or remove these ones in any moment.

No writing operations on the databases will be performed in this phase. TAP_SCHEMA data will be persisted on the databases only pressing the “Update” button in the TAP_SCHEMA editor (see [subsection 5.5](#)).

5.4 Checking for inconsistencies

When the “Edit TAP_SCHEMA” button is pressed the TAP_SCHEMA Manager will check for inconsistencies. This means that the source database structure is compared with data stored into the existing TAP_SCHEMA in order to recognize inconsistent situations like:

- TAP_SCHEMA contains schemata, tables or columns that are not contained anymore into the source database;
- a column in the TAP_SCHEMA has a different indexing or datatype compared with the source database metadata;

If some of these situations are detected the TAP_SCHEMA Manager will show a list of inconsistencies and if you click on the “Proceed” button the application will modify the TAP_SCHEMA content in order to be consistent with the database structure.

5.5 TAP_SCHEMA editing

From the TAP_SCHEMA editor interface you can add or remove schemata, tables and columns from the TAP_SCHEMA and edit their properties like utype, description and more.

On each moment you have a selected schema, table and column and you can see properties regarding that objects. Schemata and tables are displayed in a tab interface, while columns are displayed in a scrollable list.

Clicking the green buttons will open a modal dialog from which you can select elements to add into the TAP_SCHEMA.

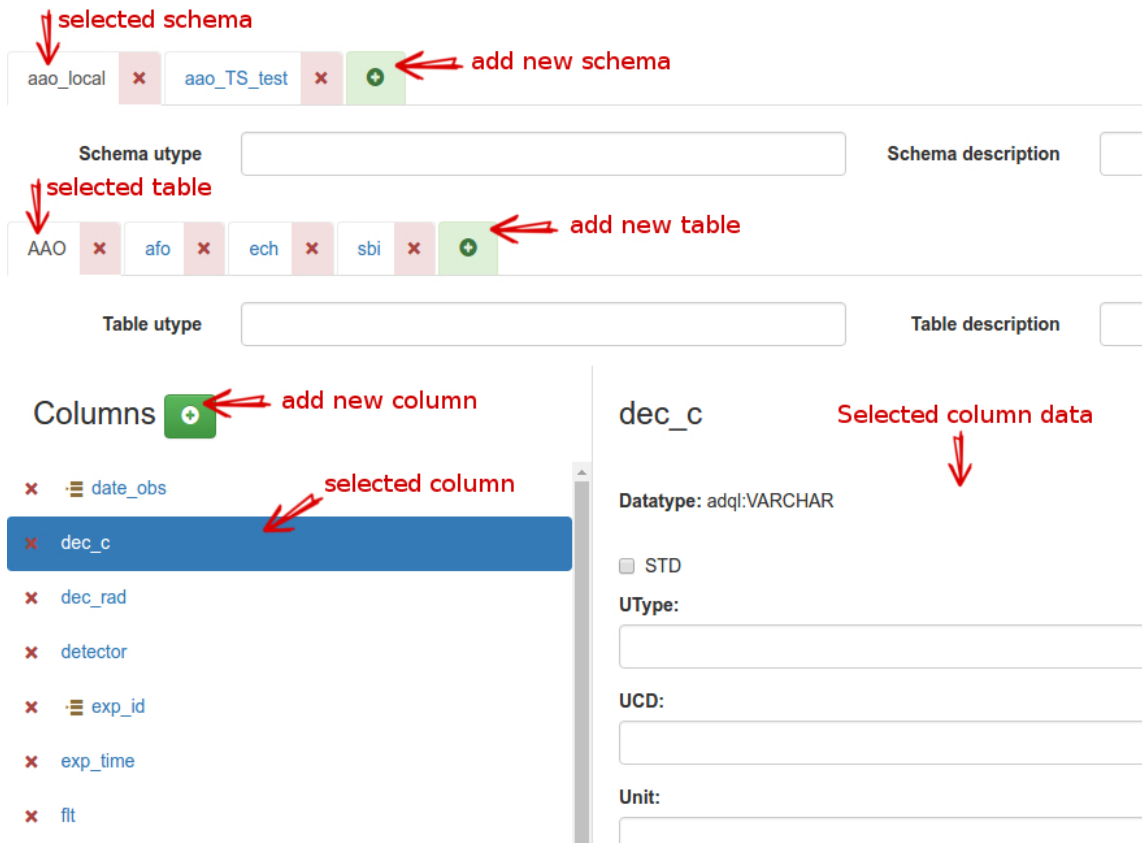


Figure 4: Editing of a TAP_SCHEMA

In the upper part of the editor page you can find some buttons (Figure 5) for performing the following actions:

- **Display update operations:** opens a modal dialog showing the list of operations that will be performed on the TAP_SCHEMA database if you press the “Update” button.
- **Update:** stores on the TAP_SCHEMA database the modifications that you have performed using the editor.
- **Reload all:** reloads the entire TAP_SCHEMA , performing the consistency check again (useful if you have changed the source database structure while the TAP_SCHEMA editor was opened).
- **Back:** returns on the “Schemata selection page”.
- **Close session:** returns on the “Credentials insertion page” (doesn’t do a complete logout from the TAP_SCHEMA Manager interface).



Figure 5: Buttons on the upper part of the page

5.5.1 UCD validation tool

Clicking on column UCD text input a modal appears and it can be used to search and validate UCDs. The user can choose:

- Search by description (Figure 7);
- Manual insertion (Figure 6);

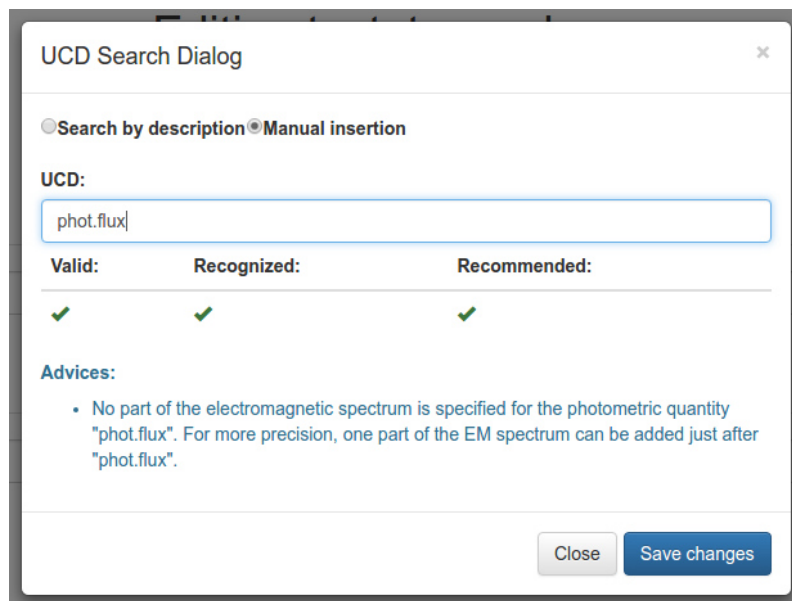


Figure 6: UCD search dialog using the UcdValidator library

The first option is based on the Strasbourg astronomical Data Center (CDS) dedicated services¹ but relies on an additional service, called UCD_REST, that wraps the SOAP CDS services into a REST architecture. The URL of this web service is specified in the configuration file property `ucd_service_url`.

The second option is based on the UcdValidator library² written by Grégory Mantelet of the Astronomisches Rechen Institut (ARI) and allows validation of manual inserted UCDs. This is useful both for validating custom UCD and in the case where the CDS services are temporary unavailable.

This part of the software is currently under development.

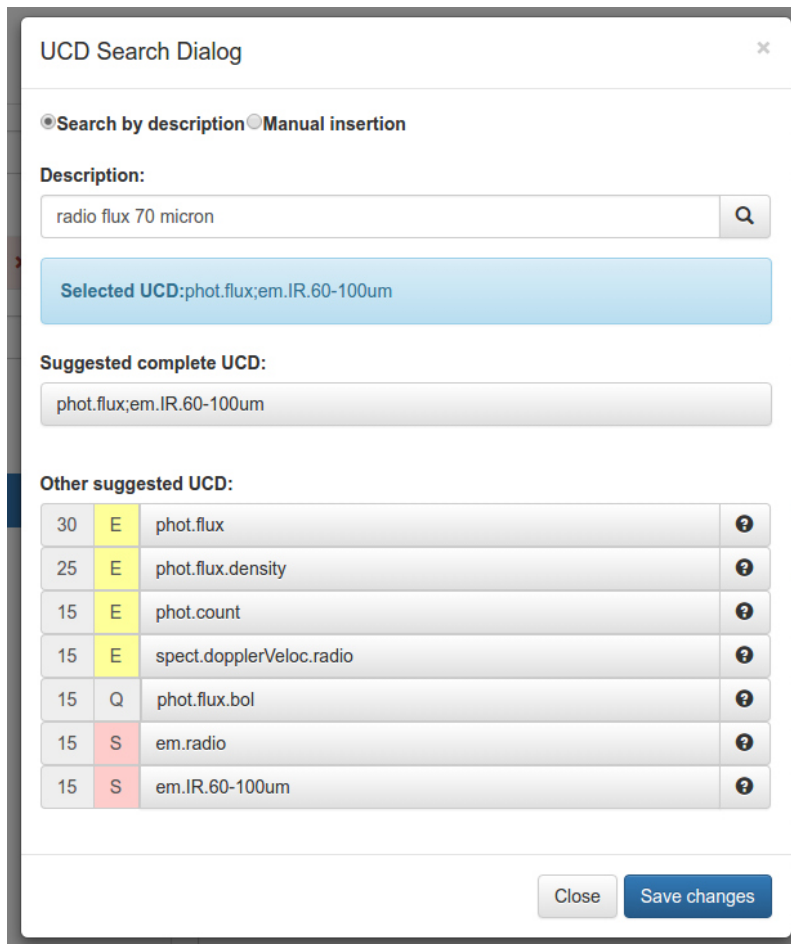


Figure 7: UCD search dialog using CDS services

¹<http://cdsweb.u-strasbg.fr/UCD/>

²<https://github.com/gmantele/ucdvalidator>