TASMAN 1.3.5 User Guide

Issue/Rev. No.:	1.3.5
Date:	November 18, 2021
Authors:	Sonia Zorba, Marco Molinaro
Approved by:	Riccardo Smareglia

Issue/Rev. No.: 1.3.5
Date: November 18, 2021
Authors: Sonia Zorba, Marco Molinaro
Approved by: Riccardo Smareglia

Contents

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.

TASMAN 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.

TASMAN 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, TASMAN can act as a client for an arbitrary numbers of database servers. In that case you have to use TASMAN with a SQL user having proper permissions (see subsection 3.1).

2 Requirements

2.1 Java

TASMAN needs Java 8 or higher.

2.2 RDMS

Currently TASMAN supports:

- MySQL (tested on version 5.6+) or MariaDB;
- PostgreSQL (tested on version 9.3+);

Note: we have done more testing on MySQL.

3 Installation

There are 2 ways for installing TASMAN:

- installing the embedded version: this consists in simply executing the self-extracting install.sh script with root privileges. Then simply run the `tasman` command for executing TASMAN.
- installing TASMAN on an existing Java EE application server: this can be done using the build.sh script contained into TASMAN git repository[1] for producing an appropriated war file to be deployed on the desired server. Follow the instruction in the README file on the repository[2].

3.1 Create a proper SQL user

TASMAN 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 TASMAN on the same machine hosting the database on which you want to work you can simply use TASMAN 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 TASMAN.

[1] `https://www.ict.inaf.it/gitlab/molinaro/tap_schema_manager`

[2] We are currently supporting GlassFish 4.1+ and Tomcat 8+, however other application servers could be easily supported including proper libraries, so if you need to run it on a specific server asks developers.

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 TASMAN 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 TASMAN 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/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 TASMAN 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.

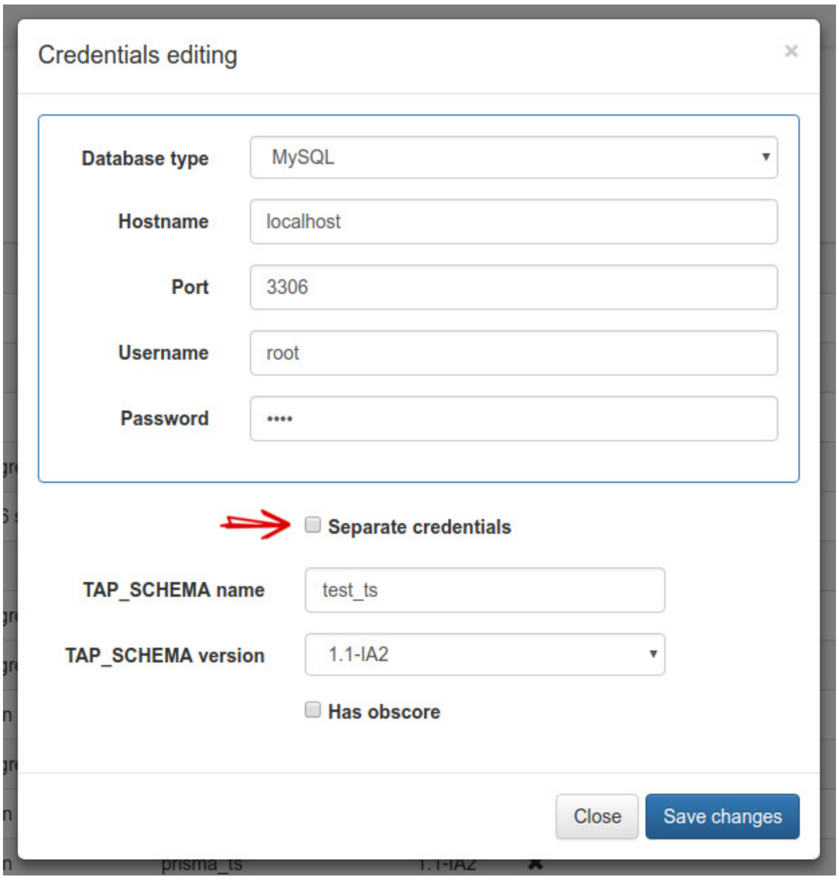
5 Usage

5.1 Login

On the first page TASMAN asks you for the password you had configured in the first setup 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 items containing information regarding database connections and TAP_SCHEMA settings.



Credentials editing

Database type: MySQL

Hostname: localhost

Port: 3306

Username: root

Password: ••••

□ Separate credentials

TAP_SCHEMA name: test_ts

TAP_SCHEMA version: 1.1-IA2

□ Has obscore

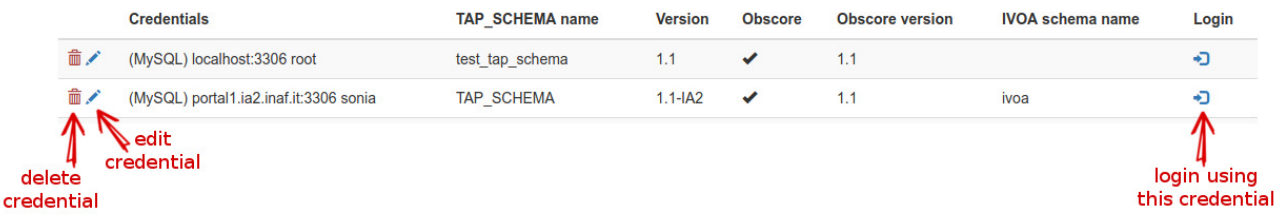
Figure 1: Credentials editing modal

For adding new credentials press the “Add new database credentials” button. A modal window will open; it contains:

- a set of inputs for specifying database connection parameters;
- a “Separate credentials” checkbox: select this if the source database server is different from the TAP_SCHEMA database server (see Figure 1);
- an input for specifying a custom TAP_SCHEMA name;
- a version for the TAP_SCHEMA (see also ...);
- a “Has obscore” checkbox: select this if you want TASMAN will generate also the ObsCore table;
- if the “Has obscore” checkbox is selected, an input for specifying a custom name for the ivoa schema (which contains the ObsCore table).

When you press the “Save changes” button, credentials will be permanently stored in the file you have specified during the installation.

Each saved credential will appear in a row of a table (Figure 2).



	Credentials	TAP_SCHEMA name	Version	Obscore	Obscore version	IVOA schema name	Login
🗑 ✎	(MySQL) localhost:3306 root	test_tap_schema	1.1	✓	1.1		➔
🗑 ✎	(MySQL) portal1.ia2.inaf.it:3306 sonia	TAP_SCHEMA	1.1-IA2	✓	1.1	ivoa	➔

Figure 2: Use a saved credential

When you press the "Login in" button from a saved credential row TASMAN will try to load the TAP_SCHEMA.

5.3 Checking for inconsistencies

During the TAP_SCHEMA loading, TASMAN 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 TASMAN 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.

5.4 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 containing a "plus symbol" (Figure 4) will open a modal dialog from which you can select elements to add into the TAP_SCHEMA.
In the upper part of the editor page you can find some buttons (Figure 3) for performing the following actions:

- **Show 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.

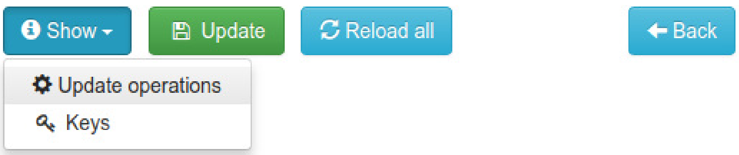
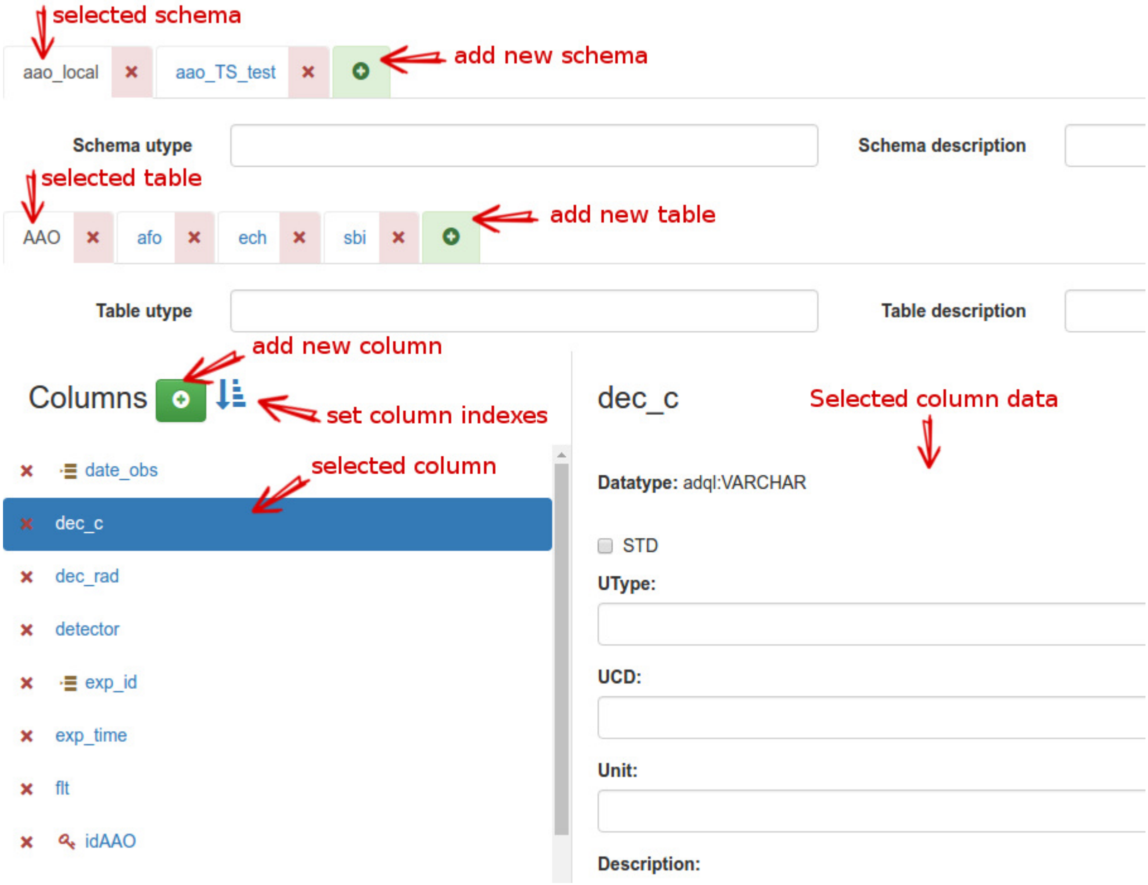


Figure 3: Buttons on the upper part of the page

- **Show keys**: opens a modal dialog showing the database keys exposed by the TAP_SCHEMA.
- **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 "Credentials insertion page".



selected schema

add new schema

selected table

add new table

add new column

set column indexes

selected column

Selected column data

Datatype: adql:VARCHAR

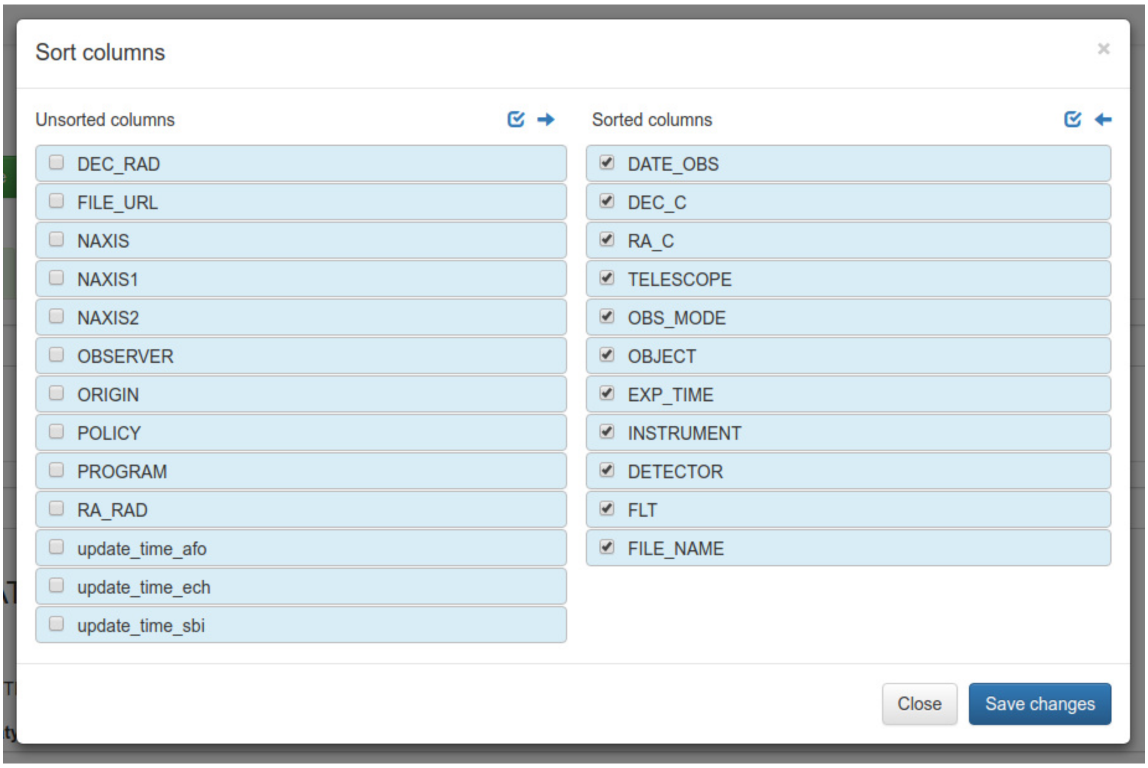
Figure 4: Editing of a TAP_SCHEMA

5.4.1 Column indexes editor

Clicking on the "sorting icon" on the top of the columns panel a modal dialog will be opened. This dialog allows to set the value of the column index properties simply sorting the columns of the selected table. The modal is split in two parts: on the left there are the unsorted columns (their column index will remain unset); on the right there are the sorted columns (first column of the list will have column index equals to 1, the index is then incremented by one on each following column).

You can sort the columns simply dragging them on the desired position.

You can also select all the columns on the left (or a subset of them) and move them to the right and vice versa.



Unsorted columns

- ☐ DEC_RAD
- ☐ FILE_URL
- ☐ NAXIS
- ☐ NAXIS1
- ☐ NAXIS2
- ☐ OBSERVER
- ☐ ORIGIN
- ☐ POLICY
- ☐ PROGRAM
- ☐ RA_RAD
- ☐ update_time_afo
- ☐ update_time_ech
- ☐ update_time_sbi

Sorted columns

- ☑ DATE_OBS
- ☑ DEC_C
- ☑ RA_C
- ☑ TELESCOPE
- ☑ OBS_MODE
- ☑ OBJECT
- ☑ EXP_TIME
- ☑ INSTRUMENT
- ☑ DETECTOR
- ☑ FLT
- ☑ FILE_NAME

Figure 5: Sorting columns for generating the column index properties

5.4.2 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 6);
- manual insertion;

The first option is based on the Strasbourg astronomical Data Center (CDS) dedicated services[3] 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 can be changed during the installation.

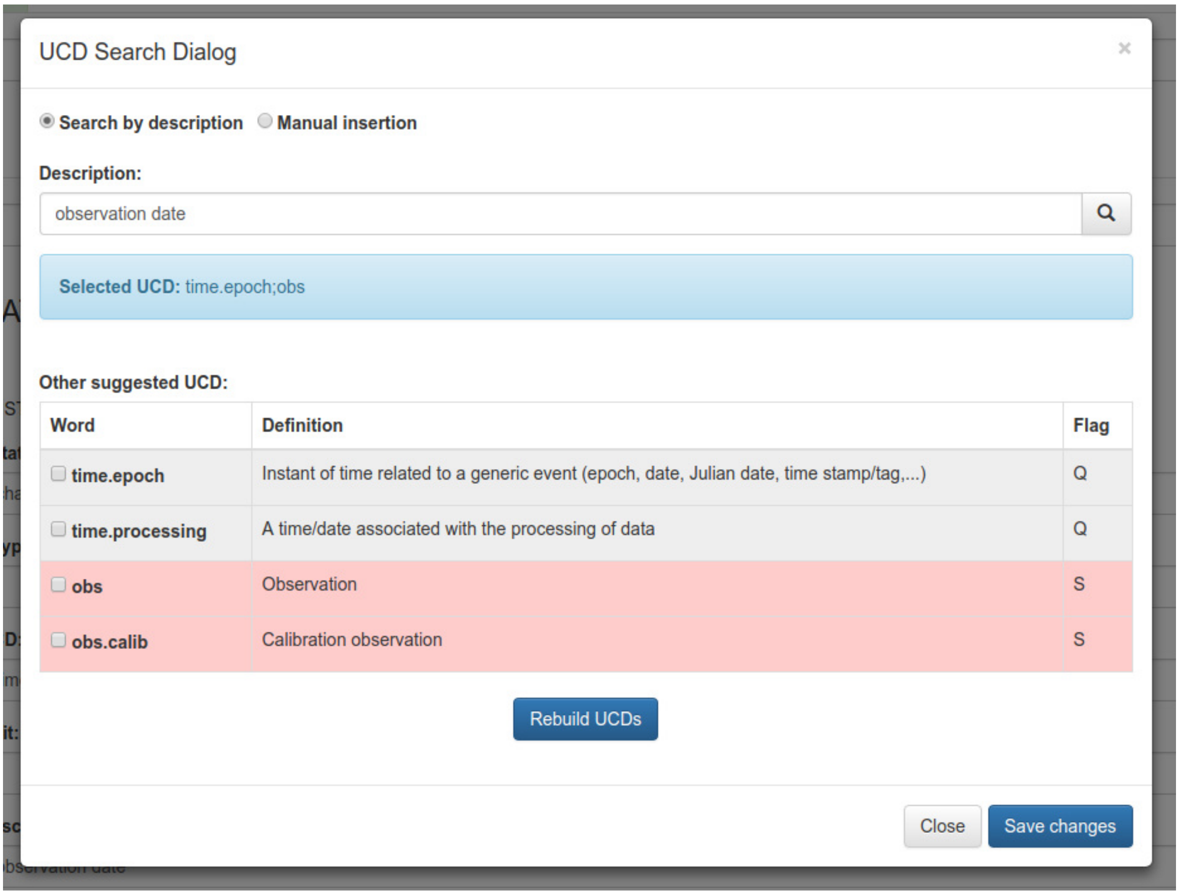
The second option is based on the UcdValidator library[4] written by Grégory Mantelet of the Astronomisches Rechen Institut (ARI) and allows validation of manual inserted UCDs. This is useful both for validating custom UCD and in the case where the CDS services are temporary unavailable.

When the UCD search dialog is opened, TASMAN uses the CDS services for searching an UCD from a textual description.

The result of the search consists in a selected UCD and a list of other available UCD words. It is possible to select some of these words and click the "Rebuild UCDs" button to obtain other valid combinations.

On the top menu bar there is a "Custom UCDs" button. It opens a modal from which the user can add some customized UCDs. It is possible to insert them using the "Manual insertion" mode on the UCD insertion dialog. Custom UCDs are shared between all TASMAN users. The admin can delete all UCDs; generic users can delete only UCDs they have created.

[3] http://cdsweb.u-strasbg.fr/UCD/
[4] https://github.com/gmantele/ucdvalidator



UCD Search Dialog

● Search by description ○ Manual insertion

Description: observation date

Selected UCD: time.epoch;obs

Other suggested UCD:

Word	Definition	Flag
□ time.epoch	Instant of time related to a generic event (epoch, date, Julian date, time stamp/tag,...)	Q
□ time.processing	A time/date associated with the processing of data	Q
□ obs	Observation	S
□ obs.calib	Calibration observation	S

Figure 6: UCD search dialog using CDS services

5.5 Users management

When TASMAN is executed for the first time, username and password for the administrator user are set up.

The administrator can use the “Users management” button on the top menu bar for adding/removing other users.

6 Advanced configuration

6.1 Creating custom TAP_SCHEMA versions

TASMAN builds TAP_SCHEMA structures dynamically using XML configuration files inside the TASMAN-core module. In this way it is easy to support multiple TAP_SCHEMA versions.

It is possible to add a custom version extending one on these XML configurations. There are 2 ways to reference the new XML files:

- Build time: the new files need to be put inside the `src/main/resources/schema_definition` folder, then their names need to be added to the list inside the `src/main/resources/core.properties` file; the project has to be compiled again;
- Execution time: the new files need to be put inside a folder named `schema_definition` inside the configuration folder; if the application is installed using the installer no additional actions are needed, otherwise

the application has to be started with an environment variable named `TASMAN_CONFIG_FOLDER` that references the configuration folder.

The custom version will appear on the TAP_SCHEMA version dropdown in the credential editing page. Operations achievable by extending the configuration:

- adding new schemata, tables and columns;
- changing a property of a column (datatype, editability, ...);
- removing tables and columns.

6.1.1 Example: extending the TAP_SCHEMA standard 1.1 version

```
<schema name="TAP_SCHEMA" version="MyCustomVersion" extends="1.1">
    <table name="columns">
        <!-- this will add a column called "my_custom_column"
         |-- to the TAP_SCHEMA columns table -->
        <column name="my_custom_column">
            <!-- data type must be in ADQL 2.0 datatype format -->
            <type>INTEGER</type>
            <!-- means the user can edit the content of this column -->
            <updatable>true</updatable>
            <!-- means this is not a standard column -->
            <standard>false</standard>
        </column>
        <!-- this will remove the arraysize column -->
        <remove name="arraysize" />
    </table>
</schema>
```