CSIRO Australia

Deploying CASDA VO Tools



25th June 2024 v1.25

CSIRO Information Management & Technology

CSIRO Astronomy & Space Science

Copyright

© Commonwealth Scientific and Industrial Research Organisation 2015-2024. To the extent permitted by law, all rights are reserved and no part of this publication covered by copyright may be reproduced or copied in any form or by any means except with the written permission of CSIRO.

Important disclaimer

CSIRO advises that the information contained in this publication comprises general statements based on scientific research. The reader is advised and needs to be aware that such information may be incomplete or unable to be used in any specific situation. No reliance or actions must therefore be made on that information without seeking prior expert professional, scientific and technical advice. To the extent permitted by law, CSIRO (including its employees and consultants) excludes all liability to any person for any consequences, including but not limited to all losses, damages, costs, expenses and any other compensation, arising directly or indirectly from using this publication (in part or in whole) and any information or material contained in it.

CSIRO is committed to providing web accessible content wherever possible. If you are having difficulties with accessing this document please contact [enquiries@csiro.au](mailto:enquiries@csiro.au).

Acknowledgments

CASDA VO Tools was developed as a part of the CSIRO Australian Square Kilometre Array Pathfinder Data Archive (CASDA) project.

CASDA is a core component in the Australian Square Kilometre Array Pathfinder (ASKAP) system. It is essentially the primary point for storing, managing, sharing and using processed ASKAP data products.

The CASDA project is a partnership between CSIRO Space & Astronomy (CS&A), CSIRO Information Management & Technology (IM&T) and the Pawsey Supercomputing Centre. It is funded by CSIRO.

CASDA project team:

Aken, Kron

Birks, Matthew

Boesen, Brendan

De Laine, Adam

Chapman, Jessica

Cheeseman, Brian

Dempsey, James

Firth, Wendy

Gosper,Heya

Greer, Bradford

Helliwell, Amanda

Heywood, Ian

Hope, Chris

Huynh, Minh

Kosmynin, Arkadi

Krishnan, Leena

Lin, XIangtan

Miller, Dan

Morrison, Dave

Mok, Wai

Moss, Sarah

Nair, Genevieve

Ophel, Margaret

Page, John

Palaniyandi, Suresh

Plackov, George

Pritchard, Jared

Pulukuri, Vivek

Ranieri, Daniel

Sanderson, Robina

Smith, Brooke

Smith, Corey

Stringfellow, Mitchell

Sullivan, Brendan

Swann, James

Trapani, Chris

Tyhouse, Rene

Van Heuzen, Rob

Whiting, Matthew

Contents

[Acknowledgments 4](#_Toc118120258)

[Contents 5](#_Toc118120259)

[1 Introduction 9](#_Toc118120260)

[2 System requirements 10](#_Toc118120261)

[3 Installing CASDA VO Tools 11](#_Toc118120262)

[3.1 Installing using the war file and Tomcat 11](#_Toc118120263)

[3.2 Building and installing CASDA VO Tools using source code 11](#_Toc118120264)

[3.3 Application log files 12](#_Toc118120265)

[3.4 Configuration Location 12](#_Toc118120266)

[4 Configuring CASDA VO Tools 13](#_Toc118120267)

[5 VO Tools configuration parameters 18](#_Toc118120268)

[5.1 General options 18](#_Toc118120269)

[5.1.1 application.base.url 18](#_Toc118120270)

[5.1.2 application.message 18](#_Toc118120271)

[5.1.3 auth.trusted.ip 18](#_Toc118120272)

[5.1.4 auth.trusted.userId 18](#_Toc118120273)

[5.1.5 build.environment 18](#_Toc118120274)

[5.1.6 connection.driverClassName 18](#_Toc118120275)

[5.1.7 connection.password 18](#_Toc118120276)

[5.1.8 connection.url 18](#_Toc118120277)

[5.1.9 connection.username 19](#_Toc118120278)

[5.1.10 default.db.schema 19](#_Toc118120279)

[5.1.11 log.timezone 19](#_Toc118120280)

[5.1.12 logo.url 19](#_Toc118120281)

[5.1.13 stylesheet.information 19](#_Toc118120282)

[5.2 SCS options 19](#_Toc118120283)

[5.2.1 scs.max.radius 19](#_Toc118120284)

[5.2.2 scs.max.records 19](#_Toc118120285)

[5.2.3 scs.outputFormat.alias 19](#_Toc118120286)

[5.2.4 scs.output.format.mime 20](#_Toc118120287)

[5.2.5 scs.test.catalog 20](#_Toc118120288)

[5.2.6 scs.test.dec 20](#_Toc118120289)

[5.2.7 scs.test.extras 20](#_Toc118120290)

[5.2.8 scs.test.ra 20](#_Toc118120291)

[5.2.9 scs.test.verbose 20](#_Toc118120292)

[5.3 TAP options 20](#_Toc118120293)

[5.3.1 tap.async.timeout 20](#_Toc118120294)

[5.3.2 tap.async.base.url 20](#_Toc118120295)

[5.3.3 tap.async.description 20](#_Toc118120296)

[5.3.4 tap.async.job.list.name 20](#_Toc118120297)

[5.3.5 tap.data.access.url 21](#_Toc118120298)

[5.3.6 tap.execution.duration.default 21](#_Toc118120299)

[5.3.7 tap.execution.duration.hard 21](#_Toc118120300)

[5.3.8 tap.job.name.prefix 21](#_Toc118120301)

[5.3.9 tap.language.description 21](#_Toc118120302)

[5.3.10 tap.language.name 21](#_Toc118120303)

[5.3.11 tap.language.version 21](#_Toc118120304)

[5.3.12 tap.max.records 21](#_Toc118120305)

[5.3.13 tap.max.running.jobs 21](#_Toc118120306)

[5.3.14 tap.output.format.alias 21](#_Toc118120307)

[5.3.15 tap.output.format.mime 21](#_Toc118120308)

[5.3.16 tap.results.dir 22](#_Toc118120309)

[5.3.17 tap.output.limit.hard 22](#_Toc118120310)

[5.3.18 tap.retention.period.default 22](#_Toc118120311)

[5.3.19 tap.retention.period.hard 22](#_Toc118120312)

[5.3.20 tap.sync.timeout 22](#_Toc118120313)

[5.3.21 tap.votable.xsl 22](#_Toc118120314)

[5.3.22 tap.examples.url 22](#_Toc118120315)

[5.4 TAP Examples 22](#_Toc118120316)

[5.4.1 examples 22](#_Toc118120317)

[5.5 TAP metadata options 25](#_Toc118120318)

[5.5.1 metadata.contactPerson 25](#_Toc118120319)

[5.5.2 metadata.copyright 25](#_Toc118120320)

[5.5.3 metadata.furtherInformation 25](#_Toc118120321)

[5.5.4 metadata.identifier 25](#_Toc118120322)

[5.5.5 metadata.instrument 25](#_Toc118120323)

[5.5.6 metadata.server 25](#_Toc118120324)

[5.5.7 metadata.servicePublisher 25](#_Toc118120325)

[5.5.8 metadata.serviceShortName 25](#_Toc118120326)

[5.5.9 metadata.serviceTitle 26](#_Toc118120327)

[5.6 Datalink options 26](#_Toc118120328)

[5.6.1 datalink.async.service.name 26](#_Toc118120329)

[5.6.2 datalink.async.service.name.internal 26](#_Toc118120330)

[5.6.3 datalink.async.service.url 26](#_Toc118120331)

[5.6.4 datalink.base.url 26](#_Toc118120332)

[5.6.5 datalink.cutout.service.name 26](#_Toc118120333)

[5.6.6 datalink.cutout.ui.service.name 26](#_Toc118120334)

[5.6.7 datalink.cutout.ui.url 26](#_Toc118120335)

[5.6.8 datalink.cutout.url 27](#_Toc118120336)

[5.6.9 datalink.download.limit.http 27](#_Toc118120337)

[5.6.10 datalink.links.url 27](#_Toc118120338)

[5.6.11 datalink.sync.service.name 27](#_Toc118120339)

[5.6.12 datalink.sync.service.name.internal 27](#_Toc118120340)

[5.6.13 datalink.sync.service.url 27](#_Toc118120341)

[5.6.14 datalink.sync.service.url.internal 28](#_Toc118120342)

[5.6.15 siap.shared.sectret.key 28](#_Toc118120343)

[5.7 SSAP options 28](#_Toc118120344)

[5.7.1 ssap.table 28](#_Toc118120345)

[5.7.2 ssap.outputLimit.hard 28](#_Toc118120346)

[5.7.3 ssap.max.records 28](#_Toc118120347)

[5.7.4 ssap.metadata.response 28](#_Toc118120348)

[5.8 SIA1 Options 29](#_Toc118120349)

[5.8.1 sia1.surveys.config.file 29](#_Toc118120350)

[5.8.2 sia1.access.url 29](#_Toc118120351)

[5.8.3 sia1.format.table 29](#_Toc118120352)

[5.8.4 sia1.outputLimit.hard 29](#_Toc118120353)

[5.8.5 sia1.max.records 29](#_Toc118120354)

[5.9 SIA1 Surveys 30](#_Toc118120355)

[5.9.1 surveys 30](#_Toc118120356)

[5.10 Schema options 30](#_Toc118120357)

[5.10.1 description 31](#_Toc118120358)

[5.10.2 utype 31](#_Toc118120359)

[5.11 Table options 31](#_Toc118120360)

[5.11.1 description 31](#_Toc118120361)

[5.11.2 description.long 31](#_Toc118120362)

[5.11.3 params 31](#_Toc118120363)

[5.11.4 release.required 31](#_Toc118120364)

[5.11.5 scs.enabled 31](#_Toc118120365)

[5.11.6 tap.schema.name 31](#_Toc118120366)

[5.11.7 tap.table.name 31](#_Toc118120367)

[5.11.8 utype 31](#_Toc118120368)

[5.12 Column options 32](#_Toc118120369)

[5.12.1 column.order 33](#_Toc118120370)

[5.12.2 description 33](#_Toc118120371)

[5.12.3 indexed 33](#_Toc118120372)

[5.12.4 principal 33](#_Toc118120373)

[5.12.5 scs.verbosity 33](#_Toc118120374)

[5.12.6 size 33](#_Toc118120375)

[5.12.7 std 33](#_Toc118120376)

[5.12.8 ucd 33](#_Toc118120377)

[5.12.9 unit 33](#_Toc118120378)

[5.12.10 utype 33](#_Toc118120379)

[5.12.11 type 34](#_Toc118120380)

[6 References 35](#_Toc118120381)

# Introduction

CASDA VO Tools is a highly customizable application developed to serve astronomy data via Virtual Observatory [1] protocols [2]. Currently Simple Cone Search (SCS) [3], Table Access (TAP) [4], Simple Image Access v2 (SIA2) [15] and Datalink [16] protocols are supported.

Each protocol has its own endpoint in VO Tools, that is, a URL where the service can be accessed by a client application.

|  |  |
| --- | --- |
| Protocol | Endpoint URL |
| SCS | <VO tools base URL>/scs |
| TAP | <VO tools base URL>/tap |
| SIA1 | <VO tools base URL>/sia1 |
| SIA2 | <VO tools base URL>/sia2 |
| Datalink | <VO tools base URL>/datalink |

Each endpoint has its own configuration parameters and database tables (and/or views) that it provides access to.

To enable access to a database table via TAP, include it in the list of tables in the TAP endpoint configuration. Then you can search these tables using ADQL queries via the TAP protocol at URLs like <VO tools base URL>/tap.

To enable access to a database table via SCS, include it in the list of tables in the SCS endpoint configuration and define the UCDs (v1.1) for the RA and Dec fields. Then you can search these tables using the SCS protocol at URLs like <VO tools base URL>/scs/<table name>, where <table name> is the name of the table to search.

Some VO clients implement cone search on top of TAP. That is, if the table columns storing RA and Dec values have the corresponding UCDs [5,6] that identify them as coordinate columns, TAP clients can recognise them as coordinates and use them in cone search.

This document provides simple step-by-step instructions for the installation of CASDA VO Tools. A basic familiarity with VO concepts is required.

# System requirements

CASDA VO Tools is a cross-platform application. It should run on any computer that has the required software installed on it. There are no special requirements for memory and disk space.

CASDA VO Tools is implemented in Java as a Web application and requires a Java servlet container, such as Apache Tomcat or Jetty, to run. Another important requirement is that the data is stored in a PostgreSQL [7] database. The reason for this is that Astronomical Data Query Language (ADQL) [8] parser and an extension for processing spherical data types (pgSphere [9]) are available for PostgreSQL, and CASDA VO Tools uses these for its implementation of TAP queries and cone search.

For downloading and deploying CASDA VO Tools war file, the following software is required:

* Apache Tomcat version 8 or later [10], or another Java servlet container;
* Java 8 JDK (future releases will move to JDK 21+) [11];
* PostgreSQL 9.4 or later [7];
* pgSphere 1.1 or later [9];

The required software packages can be downloaded using URLs provided in the references section and installed by following the steps outlined in their documentation or in tutorials that can be found on the Web.

NOTE: It is not necessary to have CASDA VO Tools and PostgreSQL with pgSphere on the same computer. CASDA VO Tools will work as long as it can access the database via JDBC.

For building CASDA VO Tools from source code, in addition to the software listed above, Java JDK (not JRE) is required. Your computer must also be connected to the Internet because the build process will have to download a considerable number of dependencies.

# Installing CASDA VO Tools

## Installing using the war file and Tomcat

1. Download VO Tools war file from [12].
2. Drop it to your Tomcat server webapps directory.
3. If your Tomcat is configured to detect and install web applications automatically, it will detect the CASDA VO Tools war file and automatically unpack it into a new web application directory.
4. If Tomcat does not unpack war files automatically, do the following (assuming that Tomcat is installed to /opt/tomcat8/ and CASDA VO Tools war file is casda\_vo\_tools.war):

#$> mkdir /opt/tomcat8/webapps/casda\_vo\_tools

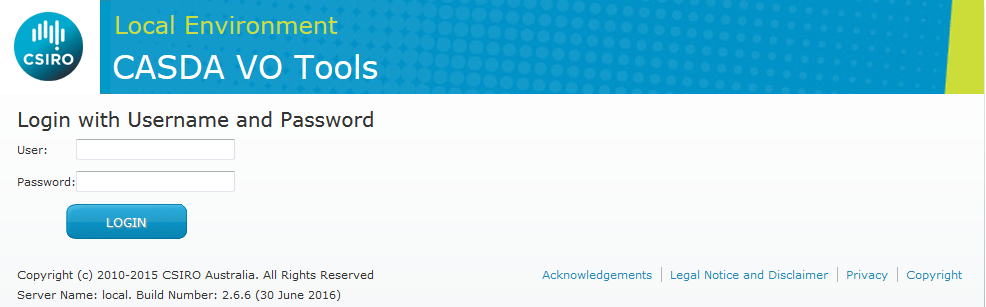
#$> mv casda\_vo\_tools.war /opt/tomcat8/webapps/casda\_vo\_tools/

#$> cd /opt/tomcat8/webapps/casda\_vo\_tools/

#$> jar –xvf casda\_vo\_tools.war

Then restart Tomcat.

Start your browser and point it to <VO tools base URL>/casda\_vo\_tools/configure/home. You should see a web page that looks like Picture 1.



Picture 1. CASDA VO Tools Login page

If you see this, it means that VO Tools is installed and running.

## Building and installing CASDA VO Tools using source code

1. Download CASDA VO Tools source code archive from [12].
2. Drop it to a directory on your hard drive, e.g. /temp.
3. Do the following:

#$> cd /temp

#$> tar -xzf casda\_vo\_tools-src.tar.gz

#$> cd /casda\_vo\_tools/

#$> ./gradlew build

If the build is successful, it will create CASDA VO Tools war file in build/libs subdirectory. Use it to install CASDA VO Tools as described in section 3.1.

## Application log files

By default, CASDA VO Tools is configured to write out log files to logs/casda\_vo\_tools.log relative to the web container’s working directory (i.e. the current directory of the container process).

A custom logging configuration can be provided at config/CasdaVoTools-log4j2.xml under the web container’s working directory. This should use the log4j 2 XML syntax.

## Configuration Location

CASDA VO Tools can be configured either via the user interface or through an external configuration file. In both cases the configuration is held in a configuration.yaml file stored in a known location on the file system. By default, this location will be the config folder under the web container’s working directory (i.e. the current directory of the container process).

This location can be customised using command line parameters to the web container. The two parameters are:

* -Dspring.config.location=file:/path/to/files
* -Dspring.config.additional-location=file:/path/to/files

The first option will set the new location of /path/to/files to be the only location that CASDA VO Tools looks for configuration files. The second option will add an additional location if you prefer to keep the default locations. Multiple options can be specified separated by commas.

# Configuring CASDA VO Tools

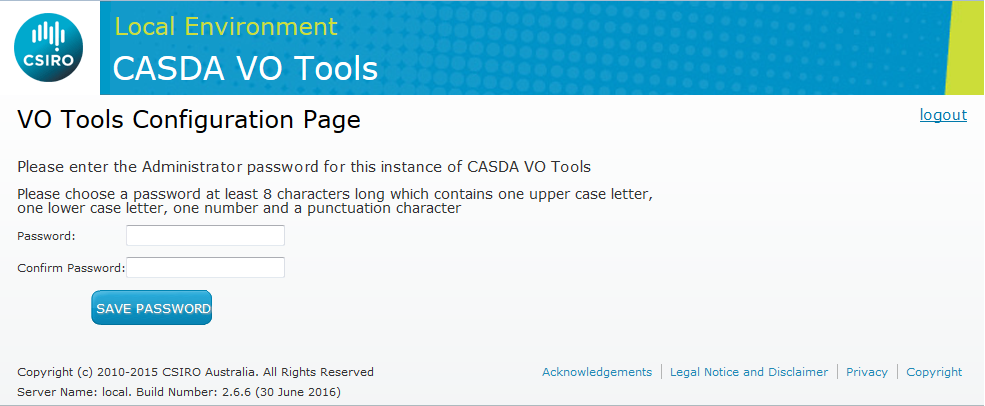
To change the configuration the administrator must be logged in. Navigating to <VO tools base URL-and-port>/admin will bring the administrator to the login screen (as shown in picture 1 above.)

When logging in for the first time, login using the following credentials:

User name: voadmin

Password: password

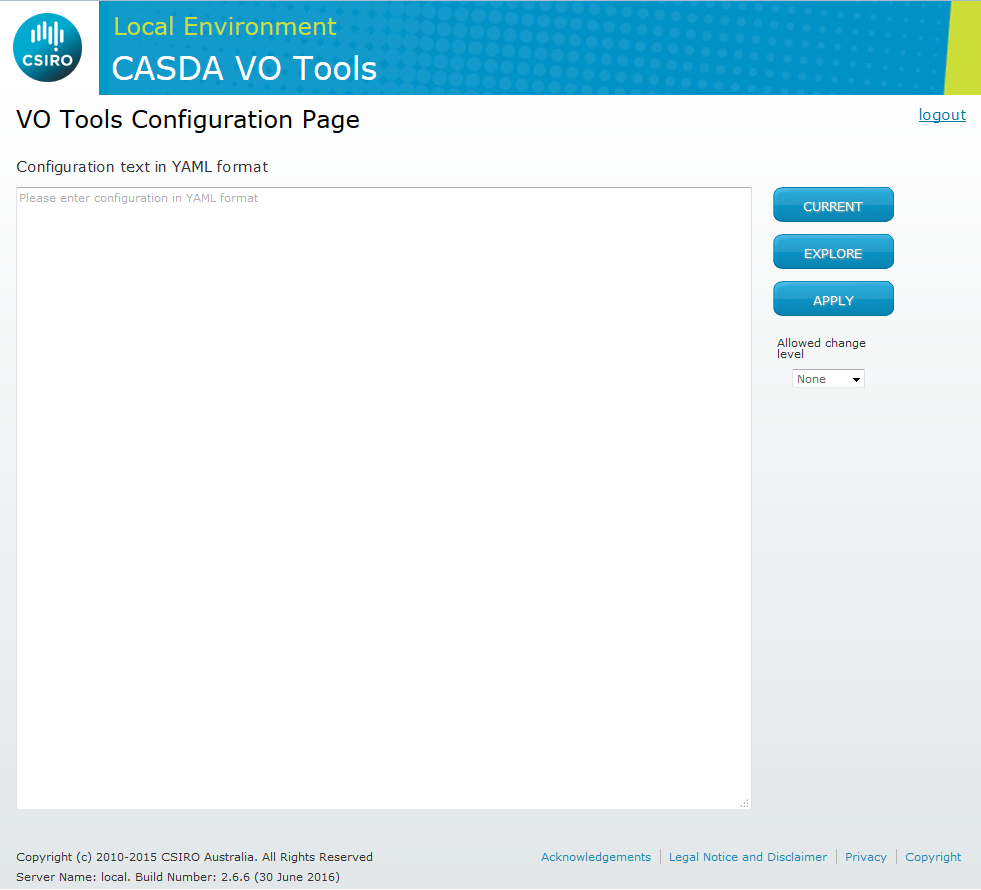
After this initial login the admin will immediately be asked to enter and confirm a new password. This password must be at least 8 characters long and include one each of the following; an uppercase letter, a lower case letter, a number, a punctuation character.



Picture 2. New password page.

This password will now be the administration password. To reset the password the administrator must enter the configuration directory (see Section 3.4) and delete the file authz.

Once the password is set the administrator will be taken to the Configuration page as shown below



Picture 3. Configuration page.

The configuration page has three large buttons:

* **CURRENT** – shows your current configuration in YAML format in the configuration window on the left.
* **EXPLORE** – reads configuration from the text window, identifies database tables mentioned in it, reads their structure, and shows complete configuration in the configuration window. It replaces missing values with placeholders looking like "<Please provide value>".
* **APPLY** – takes the submitted configuration and applies it. To do it, CASDA VO Tools connects to the database defined in the configuration and reads its structure. If this structure is different from the submitted configuration, CASDA VO Tools will try to change the structure, subject to permitted actions (see below). It also updates the configuration parameters stored in the database and stores the submitted configuration as current.

The configuration page also has a dropdown list that allows you to define the level of database changes that are permitted when applying a configuration:

* **None** - no changes to database structure are permitted.
* **Update** – permits small updates that will not cause information loss or require reindexing. Example: automatic creation of TAP metadata tables (see below).
* **Reindex** – permits updates that might require rebuilding an index. They must be performed with care because rebuilding an index can take very long time on a large database. Example: adding an index or indexed column.
* **Drop** – permits updates that may result in information loss, such as dropping columns.

The default VO Tools configuration is empty. You can use your browser and/or VO clients to connect to TAP and cone search services with base URLs **Error! Hyperlink reference not valid.** and **Error! Hyperlink reference not valid.** . However these will not have any content initially.

To switch VO Tools to your own database:

1. Click on the CURRENT button. You will see the current configuration, looking approximately like this:
2. Replace the The password to use for the database connection.

!Configuration

endPoints:

TAP:

options: !Map

tap.job.name.prefix: casda\_vo\_req\_

tap.data.access.url: http://localhost:8080//tap/jobResults/

tap.max.running.jobs: 4

tap.max.records: 20000

<… other TAP options …>

tables:

- ivoa.obscore

- casda.catalogue

SCS:

options: !Map

scs.output.format.mime: text/xml

scs.outputFormat.alias: votable

scs.max.radius: 10

<… other SCS options …>

tables:

- ivoa.obscore

options: !Map

application.base.url: http://localhost:8080/

application.message: Unconfigured install of CASDA VO Tools

log.timezone: UTC

connection.url: jdbc:postgresql://votoolsdemo.csiro.au:5432/casda

connection.username: R/OGuest

connection.password: guestpass

connection.driverClassName: org.postgresql.Driver

auth.trusted.ip: 127.0.0.1

datalink.async.service.url: <http://localhost:8080/data_access/async/>

datalink.async.service.name: [Scripted](http://localhost:8080/data_access/async/) file access via web

<… other general options …>

schemas:

ivoa: !au.csiro.casda.votools.config.SchemaConfig {}

tables:

ivoa.obscore: {},

casda.catalogue: {}

1. connection.url, The JDBC URL of the database. Example: jdbc:postgresql://votoolsdemo.csiro.au:5432/casda
2. connection.username and The name of the class implementing used JDBC driver. For PostgreSQL, it is org.postgresql.Driver.
3. connection.password with your database access parameters.
4. Remove the tables and schemas branches (below general options).
5. Replace ivoa.obscore in TAP and SCS tables lists with your own tables’ names. You can have as many of tables listed as you wish.
6. The TAP protocol requires support of special TAP metadata tables that describe all the tables that clients can access via TAP, including the metadata tables themselves. For more information, see Section 2.6 of [4]. CASDA VO Tools automatically checks for the existence of these tables and their conformance to the TAP specification and CASDA VO Tools version. If these tables do not exist or do not conform to the current CASDA VO Tools requirements, CASDA VO Tools will try to create or update them. If CASDA VO Tools does not have permissions to do it, an error message will be shown. The minimal level of permissions required to update the metadata tables is UPDATE, but, if some fields need to be removed, DROP permission is required.

**If you are installing CASDA VO Tools first time**, the system will have to create the TAP metadata tables. It creates these tables automatically in the schema specified by the [default.db.schema](#_default.db.schema) parameter. To do this, it requires the Update permitted change level selected. So, if you are installing CASDA VO Tools first time, select Update in the permitted change level select control to allow creation of the TAP metadata tables.

**If you are upgrading CASDA VO Tools**, the system checks compatibility of the existing TAP tables with the version of CASDA VO Tools you are upgrading to. It may need the Drop change level permitted if it needs to drop obsolete columns in the TAP metadata tables.

After selecting required permissions level, click on the EXPLORE button and correct configuration errors if necessary.

The system will try to read the structure of the tables that you listed in the configuration. It will then present a complete configuration with tables structure defined and placeholders for missing parameters.

1. Fill in the parameters. See the [Configuration Parameters](#_VO_Tools_configuration) section for an explanation of their meaning.
2. If you have made any changes in the configuration, select Update as permitted change level.
3. Click on the APPLY button. If you see an error message in the configuration window, something has gone wrong and you need to address the problem. You can return to the configuration by clicking on the browser’s Back button, and correct the configuration. If, after clicking on APPLY, you still see the configuration text, congratulations! Your configuration has been accepted and CASDA VO Tools should work as you’ve configured it now.

# VO Tools configuration parameters

## General options

These options define the environment in which the application is deployed.

### application.base.url

The base URL of the CASDA VO Tools application. Example: http://localhost:8080/casda\_vo\_tools

### application.message

The message displayed by CASDA VO Tools UI controller on its home page. Example: Unconfigured install of CASDA VO Tools

### auth.trusted.ip

The comma separated list of trusted IP addresses. The application reads user information from headers of requests coming from trusted hosts.

### auth.trusted.userId

The comma separated list of trusted user idents. The application reads user information from headers of requests coming from trusted users. This allows certain users to bypass restrictions on unreleased data.

*Note: these user identifiers must still originate from trusted IPs, see above property.*

### build.environment

The name of the environment this application is running on, for display in the footer of pages.

### connection.driverClassName

The name of the class implementing used JDBC driver. For PostgreSQL, it is org.postgresql.Driver.

### connection.password

The password to use for the database connection.

### connection.url

The JDBC URL of the database. Example: jdbc:postgresql://votoolsdemo.csiro.au:5432/casda

### connection.username

The user name to use for the database connection.

### default.db.schema

The name of the database schema where TAP metadata tables reside**.** CASDA VO Tools checks for existence of TAP tables in this schema and automatically creates them if required. If they exists, it checks for their compliance to the current version of VO Tools and automatically updates them if required.

### log.timezone

The time zone to use in CASDA VO Tools logs. Default: UTC

### logo.url

The URL of the image to be displayed in the header of the pages.

### stylesheet.information

The address of the css file to use to style the pages of this application.

## SCS options

These options define the Simple Cone Service and its behaviour. The values specified will be provided to users of the service via the SCS capabilities endpoint. Fields to be searched in a cone search will be identified by the UCDs pos.eq.ra;meta.main and pos.eq.dec;meta.main and thus must be present in any table to have cone search functionality provided. In addition it is recommended that an id field be identified with the UCD meta.id;meta.main .

### scs.max.radius

The maximum accepted value of search radius, degrees. Queries requesting a large radius than this value will be rejected. Default: 10

### scs.max.records

The maximum number of results records to return per request. Default: 20000

### scs.outputFormat.alias

The alias of the output format of results generated by the SCS service. Note that only votable results are supported, regardless of the alias provided. Default: votable

### scs.output.format.mime

The MIME type of the output format of results generated by the SCS service. Default: text/xml

### scs.test.catalog

The table name to use in test query in the capabilities document. Default: obscore

### scs.test.dec

The Dec value to use in test query in the capabilities document. Default: -30.45

### scs.test.extras

The extra parameters to use in test query in the capabilities document.

### scs.test.ra

The RA value to use in test query in the capabilities document. Default: 180

### scs.test.verbose

The verbosity level to use in test query in the capabilities document. Default: 1

## TAP options

These options define the Table Access Protocol service and its behaviour. This includes defining the different job limits for asynchronous and synchronous requests.

### tap.async.timeout

The timeout value for TAP database queries in asynchronous jobs, in seconds. Default: 3

### tap.async.base.url

The base URL for asynchronous TAP requests. Default: /tap

### tap.async.description

The description string for asynchronous UWS [13] service created by CASDA VO Tools. Default: UWS for CASDA

### tap.async.job.list.name

The name for asynchronous jobs list. Default: async

### tap.data.access.url

The base URL for accessing results generated by TAP. Example: http://localhost:8080/tap/jobResults/

### tap.execution.duration.default

The duration (in seconds) set by default to a job, when none is specified.

### tap.execution.duration.hard

The duration (in seconds) set by default to a job, when none is specified.

### tap.job.name.prefix

The prefix used for TAP job names. Default: casda\_vo\_req\_

### tap.language.description

The description of supported TAP query language. Should be: ADQL 2.0

### tap.language.name

The name of supported TAP query language. Should be: ADQL

### tap.language.version

The version of supported TAP query language. Should be: 2.0

### tap.max.records

The default maximum number of results records to serve per a request. Default: 20000

### tap.max.running.jobs

The maximum number of running jobs (must be > 0 to have a queue).

### tap.output.format.alias

The alias to output format of results generated by the TAP service. Default: votable

### tap.output.format.mime

The MIME type of output format of results generated by the TAP service. Default: text/xml

### tap.results.dir

The filesystem folder where TAP results files and UWS queue backups will be stored. This folder may receive large files. Relative values will be resolved relative to the current working directory. Default: temp

### tap.output.limit.hard

The upper limit to the number of records to serve per request. Default: 20000

### tap.retention.period.default

The default retention period for TAP asynchronous job results, seconds. Default: 43200 (12 hours)

### tap.retention.period.hard

The maximum retention period for TAP asynchronous job results, seconds. Default: 432000 (5 days)

### tap.sync.timeout

The timeout value for TAP database queries in synchronous jobs, in seconds. Default: 2

### tap.votable.xsl

The full URL of the XSL stylesheet used to style VOTable responses. Can be empty, None or a URL. Empty will default to the server’s votable.xsl, the text None will cause the stylesheet to be supressed and any other value will be used as the URL of the stylesheet. Default: Empty

### tap.examples.url

Url that points to an TAP examples page. When a user accesses the *“/tap/examples”* endpoint they will be redirected to the specified page. If this property is set other TAP Example Configuration methods will be ignored.

**Note.** The protocol of the TAP Service (I.e. HTTPS and HTTP) must match the protocol of the TAP examples page. Otherwise TOPCAT won’t be able to detect any configured TAP examples.

## TAP Examples

### examples

An array of examples used to build a TAP Examples page. Each example item must start with a hyphen (See below for an example). This page will be displayed from the */tap/examples* endpoint. The properties for each example are (see below for an example):

* Name: The name or title of the TAP Example.
* Query: The TAP Example query itself.
* Tables: A comma separated list of Tables the TAP Example pertains to.
* Description: A description of the TAP Example.

**Note.** The only mandatory fields, for the tap example to be discovered by VO tools such as TOPCAT, are “Name” and “Query”.

If the property “tap.examples.url” is present then any provided examples will be ignored.

!Configuration

endPoints:

TAP:

options: !Map

tap.job.name.prefix: casda\_vo\_req\_

tap.data.access.url: http://localhost:8080/tap/jobResults/

tap.max.running.jobs: 4

tap.max.records: 20000

<… other TAP options …>

tables:

- ivoa.obscore

- casda.catalogue

examples:

- name: Sky Pair Match Example

tables: casda.atlas\_continuum\_source,

query: SELECT TOP 1000 \* FROM casda.atlas\_continuum\_source AS a

JOIN casda.continuum\_component AS c

ON 1=CONTAINS(POINT('ICRS', a.ra\_deg\_cont, a.dec\_dms\_cont),

CIRCLE('ICRS', c.ra\_hms\_cont, c.dec\_dms\_cont, 5./3600.))

Description: Basic Topcat example

- name: Full Table Example

query: SELECT TOP 1000 \* FROM TAP\_SCHEMA.columns

options: !Map

scs.output.format.mime: text/xml

scs.outputFormat.alias: votable

scs.max.radius: 10

<… other SCS options …>

tables:

- ivoa.obscore

options: !Map

application.base.url: <http://localhost:8080/>

…

## TAP metadata options

These options define extra metadata which is sent back as parameters on each TAP query. They allow you to further describe your service.

### metadata.contactPerson

Who to contact about this service and description of this parameter separated by ‘|’. Example: CASDA Support &lt;atnf-datasup@csiro.au&gt;|Who to contact about this service

### metadata.copyright

The data license and description of this parameter separated by ‘|’. Example: Creative Commons Attribution 4.0 International Licence|CASDA's Data license

### metadata.furtherInformation

The link to further information on the data and usage of the service and description of this parameter separated by ‘|’. Example: http://www.atnf.csiro.au/projects/askap/index.html|Link to further information on the data and usage of the service

### metadata.identifier

The unique resource registry identifier and description of this parameter separated by ‘|’. Example: ivo://org.gavo.dc/tap|Unique resource registry identifier

### metadata.instrument

The instrument from which data originated and description of this parameter separated by ‘|’. Example: ASKAP|Instrument from which data originated

### metadata.server

The URL of the CASDA VO endpoint used and description of this parameter separated by ‘|’. Example: http://localhost:8080/casda\_vo\_tools/tap|URL of the CASDA VO endpoint used

### metadata.servicePublisher

The publisher for the TAP service and description of this parameter separated by ‘|’. Example: The CASDA team|Publisher for the TAP service

### metadata.serviceShortName

The short name of the CASDA service and description of this parameter separated by ‘|’. Example: CASDA TAP|Short name of the CASDA service

### metadata.serviceTitle

The TAP service title and description of this parameter separated by ‘|’. Example: CSIRO ASKAP Science Data Archive TAP service|TAP service title

## Datalink options

These options define the Datalink service and its behaviour. This includes defining where the link point, how they are described and what limits are placed on them. These properties allow the configuration of the aforementioned behaviours for both synchronous and asynchronous downloading.

### datalink.async.service.name

The description of the asynchronous web download link.

### datalink.async.service.name.internal

The description of the asynchronous download to an internal account. This needs only be specified where an alternate link is used for higher volume downloads. In CASDA we have a much higher download limit for requests which use a service within the Pawsey Supercomputing Centre.

### datalink.async.service.url

The URL for the asynchronous web download. E.g. <http://localhost:8080/data_access/data/async>

*Note: Leave blank to hide this service from the datalink response.*

### datalink.base.url

The base URL of the application providing the Datalink service. E.g. http://localhost:8080/

### datalink.cutout.service.name

The description of scripted cut-out service

### datalink.cutout.ui.service.name

The description of the link for the cut-out service user interface

### datalink.cutout.ui.url

The URL for the simple cut-out user interface. The request token will be appended to this url and could either be as a parameter or as part of the URL. E.g. <http://localhost:8080/cutoutui?id=> or [http://localhost:8080 /cutoutui/](http://localhost:8080%20/cutoutui/)

*Note: Leave blank to hide this service from the datalink response.*

### datalink.cutout.url

The base URL for the asynchronous cut-out service. E.g. <http://localhost:8080/cutoutui?id>=

*Note: Leave blank to hide this service from the datalink response.*

### datalink.download.limit.http

A download limit (in gigabytes) which can be placed on the synchronous web requests. Any files which are greater than this limit will result in the synchronous web request being removed from the links page and replaced by a message explaining why this option is unavailable.

### datalink.links.url

The Base URL for the datalink result page. E.g. http://localhost:8080/datalink/links

### datalink.resource

A series of properties controlling the database tables holding objects that can be advertised through datalink. Each configuration property takes an array of 3 values of the format

"database table", "id regex", "content-type"

Example: datalink.resource.image\_cube: casda.image\_cube, ^cube-[0-9]+$, application/fits

Supported properties are:

* datalink.resource.image\_cube
* datalink.resource.catalogue
* datalink.resource.spectrum
* datalink.resource.moment\_map
* datalink.resource.cubelet
* datalink.resource.evaluation
* datalink.resource.visibility
* datalink.resource.scan

### datalink.sync.service.name

The description of the synchronous web download link.

### datalink.sync.service.name.internal

The description of the synchronous download to an internal account.

### datalink.sync.service.url

The URL for the synchronous web download. The request token will be appended to this URL and could either be as a parameter or as part of the URL. E.g. [http://localhost:8080/data/sync?id={id}](http://localhost:8080/data/sync?id=%7bid%7d) or [http://localhost:8080/data/sync/{id}](http://localhost:8080/data/sync/%7bid%7d)

*Note: Leave blank to hide this service from the datalink response.*

### datalink.sync.service.url.internal

The URL for the synchronous internal download. The request token will be appended to this url and could either be as a parameter or as part of the URL. E.g. [http://localhost:8080/data/async?id={id}](http://localhost:8080/data/async?id=%7bid%7d) or [http://localhost:8080/data/async/{id}](http://localhost:8080/data/async/%7bid%7d)

### siap.shared.sectret.key

The key for encrypting the request token, the request token contains details about the request and is added as a parameter in the URL for the data access service. This key must be present in the data access service to decrypt the request token.

## SSAP options

These options define the Simple Spectral Access Protocol service and its behaviour. This includes defining if the service is active, the service metadata document and record limits.

### ssap.table

The table or view to be used for SSAP queries. If this key is not present then the SSAP service will be inactive.

### ssap.outputLimit.hard

The upper limit of the client provided MAXREC parameter. This is the maximum number of records the client can request.

### ssap.max.records

The default number of records to be returned, when no MAXREC value is provided by the client.

### ssap.metadata.response

The optional name of the file holding the SSAP metadata response. If this file is not specified and the SSAP service is active a default metadata record will be returned.

## SIA1 Options

CASDA VO Tools supports both Simple Image Access v1 and v2. SIA2 does not require any config as it automatically activates if there is an obscore table defined in TAP. SIA1 does require configuring both for supported services and to link to the data access service.

### sia1.surveys.config.file

The Location of the XML config file describing the surveys enabled for SIA1 access. An example XML config file is shown below:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Surveys>

<Survey>

<Code>RACS-Low</Code>

<Name>RACS Low Epoch 1</Name>

<Description>RACS DR1 image set</Description>

<WhereClause>(dataproduct\_type = 'cube' and calib\_level = 3 and filename LIKE 'RACS%DR1%A.fits')</WhereClause>

</Survey>

</Surveys>

### sia1.access.url

The URL to produce a cutout in the required format. The url should include {obs\_publisher\_did} , {access\_format} , {pos} and {size} placeholders to allow the cutout parameters to be supplied in the call. The call should respond to a GET query with the requested data.

### sia1.format.table

The table or view holding the list of supported SIA1 formats. It must be defined as a table in the TAP metadata. It must contain at least the columns content\_type and pix\_flags, which should contain ‘image/fits’, ‘C’ and ‘image/png’, ‘V’.

### sia1.outputLimit.hard

The upper limit of the client provided MAXREC parameter. This is the maximum number of records the client can request.

### sia1.max.records

The default number of records to be returned, when no MAXREC value is provided by the client.

## SIA1 Surveys

The CASDA SIA1 service is based around surveys, where a caller can find the best data at their point of interest from a specific survey. Multiple surveys are supported, and they can be defined either in the yaml file, or in a standalone surveys XML file. In the YAML file they should be defined under a surveys entry under the SIA1 endpoint.

### surveys

An array of surveys used in SIA1 queries. Each survey item must start with a hyphen (See below for an example). The surveys are accessible in the custom */sia1/surveys* endpoint. The properties for each survey are (see below for an example):

* Code: A unique code for the survey which is passed to the SIA1 endpoint to specify the survey
* Name: The human readable short name for the survey.
* whereClause: The where clause fragment to be used in querying obscore and restricting the results to just this survey.
* Description: A longer description of the survey.

!Configuration

endPoints:

SIA1:

surveys:

- code: RACS-Low

name: RACS Low Epoch 1

whereClause: (dataproduct\_type = 'cube' and calib\_level = 3

and filename LIKE 'RACS%DR1%A.fits')

Description: RACS DR1 image set from Hale et al. 2021

- code: Survey2

name: A second survey

whereClause: (dataproduct\_type = 'cube' and obs\_collection = 'Survey2')

Description: Second survey of the southern sky

## Schema options

A schema entry is required for each logical schema to be made available via TAP. These schema need not be related to database schema but are instead a way to group tables in a client interface such as TOPCAT.

### description

The brief description of the schema

### utype

The UTYPE [14] if the schema corresponds to a data model

## Table options

A table entry is required for each database table to be made available via TAP. Note that the name of the table is the actual fully qualified database table name. The table name and table schema name used when querying the table in TAP are defined by the tap.table.name and tap.schema.name options.

### description

The brief description of the table

### description.long

The long description of the table

### params

The additional parameters of the table

### release.required

Should unauthorised access to this table be restricted to released data, i.e. has a populated released\_date column. The value is 0 or 1.

### scs.enabled

Should this table be queryable using the Simple Cone Search service. The value is 0 or 1.

### tap.schema.name

Logical schema name used to access this table via TAP.

### tap.table.name

Logical table name used to access this table via TAP.

### utype

The UTYPE if the table corresponds to a data model. This is meant to express the role of the table in the context of an external data model [14].

## Column options

An example column definition is shown below. Many of these options will be autocompleted by the explore function in CASDA VO Tools. Any options not relevant to the column can be left out of the definition. Each possible option is described below also.

schemas:

psrda: !au.csiro.casda.votools.config.SchemaConfig

options: !Map

description: ATNF pulsar data archive

tables:

psrda.pulsar\_obs:

columns: !Map

ra\_angle: !Column

options: !Map

column\_order: 7

size: 8

description: Right Ascension

db\_column\_name: ra\_angle

unit: deg

ucd: pos.eq.ra;meta.main

scs\_verbosity: 1

principal: 1

type: double precision

dec\_angle: !Column

options: !Map

column\_order: 8

size: 8

description: Declination

db\_column\_name: dec\_angle

unit: deg

ucd: pos.eq.dec;meta.main

scs\_verbosity: 1

principal: 1

type: double precision

### column.order

The number of this column in the ordered set of columns of this table.

### description

The short description of the column

### indexed

1 if the column is indexed, else 0.

### principal

1 if this item is of main importance, and for instance is recommended in a select or should be shown in first priority in a query response, else 0.

### scs.verbosity

The lowest verbosity level (1, 2, 3) at which this column should be included in a cone search result. Set to 0 to hide the column from SCS.

### size

The size of this column, if not empty, in a database record, in bytes

### std

1 if this column is defined by some IVOA standard as opposed to a customized metadata defined by a specific service, else 0.

### ucd

The Unified Content Descriptor [5,6]. Note that this should be a UCD v1.1 value as used for TAP. When the column is presented as part of a cone search result, the UCD will be translated to UCD v1 if required. UCDs which will be translated are meta.id;meta.main, pos.eq.ra;meta.main and pos.eq.dec;meta.main . The UCD is also used by CASDA VO Tools to identify the fields to be used for the cone search.

### unit

The units in which the column value is measured, if applicable

### utype

The UTYPE if the column corresponds to a data model. This is meant to express the role of the column in the context of an external data model [14].

### type

Note that type should be specified as a sibling to the options under a column. It should not be provided as an option.

The type should be the database type of the column. The system will translate this to the appropriate TAP type or VOTable type as needed.

# References

1. Introduction to VO Concepts. <http://www.ivoa.net/deployers/intro_to_vo_concepts.html>
2. VO Documents and Standards. <http://www.ivoa.net/documents/index.html>
3. Simple Cone Search Protocol. <http://www.ivoa.net/documents/latest/ConeSearch.html>
4. Table Access Protocol. <http://www.ivoa.net/documents/TAP/>
5. An IVOA Standard for Unified Content Descriptors. <http://www.ivoa.net/documents/latest/UCD.html>
6. UCD Resolver. <http://dc.zah.uni-heidelberg.de/ucds/ui/ui/form>
7. PostgreSQL. <http://www.postgresql.org/>
8. IVOA Astronomical Data Query Language. <http://www.ivoa.net/documents/latest/ADQL.html>
9. pgSphere. <http://pgsphere.projects.pgfoundry.org/>
10. Apache Tomcat. <http://tomcat.apache.org/>
11. Java. <https://java.com/en/download/>
12. CASDA VO Tools. <https://data.csiro.au/dap/search?q=casda+vo+tools> or <https://github.com/csiro-rds/casda_vo_tools>
13. UWS Library 3.0. <http://cds.u-strasbg.fr/resources/doku.php?id=uwslib>
14. Utypes definition and syntax. <http://wiki.ivoa.net/bin/view/IVOA/Utypes>
15. Simple Image Access protocol v2.0. <http://www.ivoa.net/documents/SIA/20151223/>
16. DataLink v1.0. <http://www.ivoa.net/documents/DataLink/20150617/>

|  |  |  |
| --- | --- | --- |
|  | | |
| CONTACT US  t 1300 363 400  +61 3 9545 2176  e enquiries@csiro.au  w www.csiro.au  At CSIRO we shape the future  We do this by using science to solve real issues. Our research makes a difference to industry, people and the planet.  As Australia’s national science agency we’ve been pushing the edge of what’s possible for over 85 years. Today we have more than 5,000 talented people working out of 50-plus centres in Australia and internationally. Our people work closely with industry and communities to leave a lasting legacy. Collectively, our innovation and excellence places us in the top ten applied research agencies in the world.  WE ASK, WE SEEK AND WE SOLVE |  | For further information  CSIRO Information Management and Technology  James Dempsey t +61 2 6214 2912  e James.Dempsey@csiro.au  w www.csiro.au  CSIRO Astronomy and Space Science  Minh Huynh t +61 8 6436 8696  e Minh.Huynh@csiro.au  w www.atnf.csiro.au |