

# Chimera Database Controller

Brandon Gilfus

06-18 August 2009

## Setup

### Installation

The database is a part of Chimera and will be installed when you install Chimera. The database functionality requires that the following modules be installed (these should also have been installed with Chimera): Elixir version 0.6 or higher and SQLAlchemy version 0.5 or higher.

### Basic configuration

Once Chimera is installed, the database will need to be configured. Edit the `$HOME/.chimera/chimera.config` file to include the following lines (For more information, see [http://code.google.com/p/chimera/wiki/Chimera\\_config](http://code.google.com/p/chimera/wiki/Chimera_config) for more information about configuring Chimera). Only the first two lines are required, but you can also add and edit the other lines to customize the behavior of the database. Explanation will be interspersed with the configuration lines.

```
- type: Database
  name: db
```

These first two lines are mandatory to enable the database. The first line specifies that you're using a database, and the second line names it. Note that if you can use any name for the database that you want.

### Additional configuration (optional)

```
database_file: photometry.db
```

This line specifies the file where the contents of the database will be stored. The displayed file name (as with all the values displayed in this listing) is the default.

```
auto_add_landolt: True
```

When the database is first created, it optionally downloads a catalog of Landolt stars from the Internet, depending on whether or not this value is set.

```
auto_catalog_enabled: True
auto_catalog_name: (Auto)
```

When the auto catalog is enabled, it serves as a collection of stars entered into the database that could not be found in any catalogs already stored in the database. You can choose to specify an alternative name for this catalog if you choose.

```
auto_add_simbad: False
simbad_objtypes: *iC,~V*
simbad_entries: simbad_sources.txt
```

In addition to the Landolt download, the database can optionally download any number of fields from the Simbad online catalog service. These three settings enable the download, specify which types of objects to download from Simbad, and the name of a file containing a list of fields to download (the format of this file will be covered later).

```
auto_add_on_expose: False
camera: /Camera/0
auto_add_on_photometry: False
photometry: /Photometry/0
auto_add_on_reduction: False
reduction: /Reduction/0
```

The database is capable of monitoring several other parts of Chimera to watch for potential data updates. These settings let you configure the database to automatically add an exposure to the database as soon as the CCD has finished exposing, an image has been reduced, photometry has been completed for an image, or some combination of the above. (Note: At the time this was written, reduction was not fully operational, and the automatic update after reduction has not been fully implemented.) If you enable the database to respond to these events, you will also need to provide the full name of the appropriate objects so that the database can find them and connect to them.

```
get_star_ra_tolerance: 12
get_star_dec_tolerance: 6
```

These settings allow you to fine tune how the database matches point sources from photometry with existing stars in the database. Both settings are measured in degree arc seconds. In general, higher tolerances lead to more false positive matches and lower tolerances lead to more false negative matches, so these should be based on the amount of error in the RA and Dec of the stars in the images to be put into the database.

### ***Importing Simbad Fields (Optional)***

To import Simbad fields at initial setup (you can also import them later using the `chimera-db script`), create a text file with lines having any of the following structures:

```
Field-name;radius;catalogs;object-types
Field-name;radius;catalogs
Field-name;radius
Field-name
```

The field name is the name of an object listed in Simbad that defines the center of the field. The radius measures the size of the field in arc minutes (the default is 10). The catalogs are a means of organizing stars in the database. You can store the stars in one or more catalogs (comma-separated). If the catalog is not specified, the default is one called "Simbad". Finally, you can specify

the object types to filter by on a per-field basis. If specified, this overrides the `simbad_objtypes` setting from the configuration.

You can specify the object types as a comma-delimited list of types to include and types to exclude (preceded by a “~”). These types are the 3-letter types mentioned at <http://simbad.u-strasbg.fr/simbad/sim-display?data=otypes>.

## ***Startup***

The first time that the Chimera server is run with the database configured, the database will set itself up automatically. (Similarly, in the event that the database file is deleted, a new database file will be automatically created the next time the Chimera server is started.)

## ***Contents of the Database***

The majority of the database is dedicated to collecting and organizing data collected from photometry.

There is a table of stars, consisting essentially of right ascension/declination coordinates, some sort of name, and possibly other information imported from catalogs. (For example, apparent magnitudes in U, B, V, R, and I are provided for the Landolt stars.) Stars can be attached to one or more catalogs, depending on the source of the data.

To organize the stars, there is a table of catalogs, which are simply named lists of stars.

There is also a table of exposures, consisting of the exposure’s FITS file path, selected data from the FITS header, and a list of time-stamped annotations including the processing tasks through which the exposure image has gone (such as storage, point verification, reduction, and photometry).

Another table contains the point sources identified by photometry for all of the exposures. This table is designed to be quickly filtered by the star to which the point source corresponds or the exposure in which the point source was identified. For each point source, the coordinates are recorded, as are the FWHM, flux, instrumental magnitude, and airmass.

Additionally, an additional table in the database is dedicated to storing extinction data, including the first- and second-order extinction coefficients over a given time range and filter.

## ***Using the Database***

Some database functions can be handled automatically, using either the monitoring mechanism described in the configuration instructions or the Chimera scheduler. However, there are also two ways to work with the database manually. The simplest way is to use the `chimera-db` script, which provides many basic features for adding content, querying, and maintaining the database. Slightly more complicated is building a Python script that accesses the database through Chimera and uses its API functions directly, a discussion of which is beyond the scope of this document.

## ***Adding Content to the Database***

Exposures can be added automatically using the `chimera-cam` or photometry scripts with the appropriate settings enabled, as discussed above. The Chimera scheduler will also be able to add content to the database.

There are several options available in `chimera-db` to let you add content to the database.

```
chimera-db --add --ra=RA --dec=DEC
```

This command adds a star without a name to the database. (Note: `chimera-db` expects RA to be entered in HH:MM:SS and Dec to be entered in DD:MM:SS.)

```
chimera-db --add --star=STARNAME --ra=RA --dec=DEC
```

This command adds a star with the specified name to the database.

```
chimera-db --add --catalog=CATALOG
```

The command adds an empty catalog with the specified name to the database.

```
chimera-db --add --catalog=CATALOG --star=STAR
chimera-db --add --catalog=CATALOG --ra=RA --dec=DEC
```

These commands add the specified star to the specified catalog.

```
chimera-db --add --exposure=FILEPATH
```

This command adds an image from the specified file path to the database, automatically importing data from its FITS header.

```
chimera-db --add --exposure=FILEPATH --note=NOTE
```

This command adds an annotation to the specified image, with the timestamp set to the current date and time.

```
chimera-db --simbad --star=STARNAME --radius=RADIUS --catalog=CATALOGS --types=TYPES
chimera-db --simbad --ra=RA --dec=DEC --radius=RADIUS --catalog=CATALOGS --types=TYPES
```

These commands let you import a Simbad field into the database manually. The parameters are similar to those used in the automatic Simbad import from startup mentioned in “Importing Simbad Fields” under configuration. As there, the radius, catalog, and types parameters are optional.

## Querying the Database

Some Chimera modules, such as extinction, query the database automatically through its API. Users have two other ways to query the data in the database. The most basic method is using the query options in `chimera-db`, and a more advanced method is using view scripts, discussed later in “Producing Tables and Plots Using the Database”.

The `chimera-db` script includes several query options designed to give an overview of the contents in the database.

```
chimera-db --query --stars
```

This command displays a list of all the stars in the database. (Tip: For any command that displays lists, the `--count` parameter can be added to omit the lists, displaying only the count of items in the list.)

```
chimera-db --query --star=STARNAME
chimera-db --query --ra=RA --dec=DEC
```

These commands query for a single star and display all available information about that star if it is found, including its name, RA and Dec, catalogs, and stored apparent magnitudes.

```
chimera-db --query --catalogs
```

This command displays a list of all the star catalogs in the database.

```
chimera-db --query --catalog=CATALOG
```

This command displays a list of all the stars in the specified catalog. (Tip: The `--stars` and `--catalog` options also let you specify a `--has-exposures` option that limits the list to stars which have been found in exposures in the database by photometry.)

```
chimera-db --query --exposures
```

This command displays a list of all the exposures in the database.

```
chimera-db --query --exposure=FILEPATH
```

This command queries for a single exposure and displays all available information about that exposure if it is found, including its FITS header data, a list of annotations, and a list of stars found in the image by photometry.

```
chimera-db --query --exposures --stars
```

This command displays all the stars found in all the images by photometry, organized by the image in which they are found.

```
chimera-db --query --star=STARNAME --exposure=FILEPATH
chimera-db --query --ra=RA --dec=DEC --exposure=FILEPATH
```

These commands query for a single star found in an image by photometry, and, if the star is found, displays the photometry information for that star in that image, including RA and Dec, both expected and detected, altitude and azimuth, selected exposure information, FWHM, flux, instrumental magnitude, and airmass.

## ***Removing Content from the Database***

There is only minimal support for removing content from the database in a few scenarios. Most of these are available in chimera-db.

```
chimera-db --remove --catalog=CATALOG --star=STAR
chimera-db --remove --catalog=CATALOG --ra=RA --dec=DEC
```

These commands remove a star from a catalog.

```
chimera-db --remove --catalog=CATALOG
```

This command removes a catalog from the database.

```
chimera-db --remove --exposure=FILEPATH --note=NOTE
```

This command removes an annotation from an exposure in the database.

```
chimera-db --remove --star=STARNAME --exposure=FILEPATH
chimera-db --remove --ra=RA --dec=DEC --exposure=FILEPATH
```

These commands remove a star found by photometry from an image.

## ***Backing up the Database***

The database features the ability to back itself up to an XML file and restore itself from that file. The chimera-db script features two commands supporting this feature.

```
chimera-db --backup=FILEPATH
chimera-db --restore=FILEPATH
```

## ***More Notes about chimera-db***

The chimera-db script includes abbreviated names for many of its options:

<b>Parameter</b>	<b>Short version</b>	<b>Parameter</b>	<b>Short version</b>
--star	-s	--add	-A
--ra	-r	--query	-Q
--dec	-d	--remove	-D
--catalog	-c	--simbad	-S
--exposure	-e	--backup	-B
--note	-n	--restore	-R
--count	-N	--view	-V

## ***Producing Tables and Plots Using the Database***

The database includes a module called dataview which can be used to generate more fine-tuned queries than is possible with just chimera-db and store the results in a tab-delimited file, a table on an HTML page, or a plot in a PNG image. It supports commands that can be accessed through Python and Chimera, as well as its own simple scripting language.

### ***Scripting commands***

The dataview scripting language has just five commands, outlined below:

`base TABLE`

This command specifies which table to use as the basis for the query (tables are listed below).

`batch NAME, FIELD, OPERATION, CRITERIA`

This command creates a batch of data based on a filter of a field with the given operation and criteria. Batches are used by dataview to display sets of data with different filters parallel to each other (for example, plotting different kinds of data on the same axes or displaying a table with data collected under different filters).

`filter FIELD, OPERATION, CRITERIA`

This command applies a filter to either a batch (if a batch is specified in the field) or to all batches (if no batch is specified). Filter operations are discussed at length below.

`column NAME, FIELD`

This command creates a column in the results table with the specified heading and contents from the queried results in the given field. (Note: If you want to create a plot, the first column represents the x-values and the second column represents the y-values. Additional plots can be made on the same axes by adding additional columns, alternating between x and y-values.)

`output TYPE, TITLE, FILEPATH`

This command runs the query and saves the results in the specified file. The result will also have the given title on it. The types of output are as follows: `tabs`, `html`, and `plot`. These types produce tab-delimited text files, HTML pages, and PNG images, respectively. This command can be run multiple times to create different forms of output, and additional commands can be run between outputs.

## **Tables and fields**

There are three tables supported by dataview: STAR, EXPOSURE, and EXPSTAR. Note that queries based on EXPSTAR can access fields in all three tables, but queries based on the other tables will only have the one table accessible.

When a command requires a field, it should be specified in one of the following formats:

```
BATCH . TABLE . FIELDNAME
TABLE . FIELDNAME
BATCH . FIELDNAME
FIELDNAME
```

If the batch is not specified, it is assumed to be the default (global) batch. If the table is not specified, it is assumed to be EXPSTAR.

Each table has its own set of fields. Most fields can be used for both filtering and displaying. Below is a list of all the available fields, grouped by table.

<b>STAR</b>	<b>EXPOSURE</b>	<b>EXPSTAR</b>
RA	RA	STAR <sup>1</sup>
DEC	DEC	EXPOSURE <sup>1</sup>
NAME	ALT	ALT
CATALOGS <sup>2</sup>	AZ	AZ
MAGNITUDES <sup>3</sup>	OBSDATE	FWHM
MAGFILTERS <sup>2</sup>	LST	FLUX
EXPSTARCOUNT <sup>4</sup>	LATITUDE	MAGNITUDE
	LONGITUDE	AIRMASS
	FILTER	
	OBSERVER	
	IMAGETYPE	
	EXPTIME	
	FILENAME	
	EXPSTARCOUNT <sup>4</sup>	
	NOTES <sup>2</sup>	

<sup>1</sup> These fields can be used for displaying but not for filtering.

<sup>2</sup> These fields define lists. When filtering, it only checks if the criteria matches any item in the list, and, when displaying, it displays the whole list.

<sup>3</sup> This field is unique. When filtering, it checks if the criteria matches against all the magnitudes of the star, and, when displaying, it displays both the filters and their corresponding magnitudes.

<sup>4</sup> These fields only support one filter: whether or not there are any exposure-stars (expressed as a minimum of 1).

## ***Filter operations***

There are five operations available for the filter and batch commands.

`exact` - requires an exact match in that field

`min` - specifies a minimum value for the field

`max` - specifies a maximum value for the field

`range` - specifies both a minimum and a maximum value for the field (this operation takes two values for criteria)

`error` - specifies a value and the error around that value that will be accepted

## ***Parameters***

You can enter parameters to be replaced when the script is run. These parameters start with %.

## ***White space and comments***

You can use as much white space in the script as you want. You can also enter comment lines, starting with #.

## ***Running dataview scripts***

Dataview can be run using `chimera-db`. The command is as follows:

```
chimera-db --view=FILEPATH --params=PARAMETERS
```

Parameters can be optionally sent using `--params`. The format of `PARAMETERS` is as follows: `NAME:VALUE,NAME:VALUE,...` where `NAME` is the name used in the script without the % and `VALUE` is the value to be used in place of the parameter.