The CIWS Reference Catalogue

R.I. INAF OATo Torino n. xx, 2014

Custodian:	Name:	L. Nicastro	Signature:		Date:
Prepared by:	Name:	R. Smart, R. Morbidelli, L. Nicastro	Signature:		Date: 6/5/14
Reviewed by:	Name:	A. Bulgarelli	Signature:		Date:
Approved by:		M. Trifoglio	Signature:		Date:

All information contained in this document is property of **INAF.** All rights reserved.

AUTHOR LIST

Richard Smart, Roberto Morbidelli	INAF/OA Torino, Italy
Luciano Nicastro	INAF/IASF Bologna, Italy

DISTRIBUTION LIST

CIWS collaboration

All information contained in this document is property of **INAF.** All rights reserved.

DOCUMENT HISTORY

Version	**Date**	**Modification**
1.0	30 April 2014	First draft
1.1	6 May 2014	Review and additions

All information contained in this document is property of **INAF.** All rights reserved.

TABLE OF CONTENTS

All information contained in this document is property of **INAF.** All rights reserved.

1. Introduction

In this document we describe the CIWS Reference Catalogue (CIWS Cat) and the database access software produced for the Customizable Instrument Workstation Software Framework (CIWS-FW). CIWS Cat represents a customized combination of three different Celestial Catalogues selected among those developed by OATo (TOCats): GSC2.3, 2MASS and WISE. The software API developed to access all the MySQL based catalogues makes use of the MCS software library developed and maintained at IASF-Bo. The current version of the C++ API is working but does not implement yet all the capabilities (see Doxygen documentation at ross.iasfbo.inaf.it/CiwsCats/). On the other hand a stand-alone tool is provided that can query a user's specified region of the sky and return the information about the selected objects in various format (`getCIWScat`). This tool performs a direct access to a local or remote MySQL table containing the catalogue that were indexed using the HTM and HEALPix through the DIF facility.

All information contained in this document is property of **INAF.** All rights reserved.

2. History and Philosophy

The TOCats is a collection of astronomical catalogues stored into a relational database and is the fruit of a 10 year collaboration between OATo and IASF-Bo. As such much of the structure, decisions on internal storage and philosophy is based on the available hardware and software 10 years ago. The structure and content is also driven strongly by the primary scientific aims of the database which were to allow all sky quality assurance of large ground based catalogues - in particular for the Second Guide Star Catalog (GSCII) project, and, to allow quick access to relatively complete, multi band, object lists for rapid Gamma Ray Burst followup - in particular for the Rapid Eye Mount (REM) telescope at La Silla. If we had to start again this database, it would have a different aspect and philosophy, that would take advantage of the much richer hardware and software options available today. Nevertheless we believe we still have a useful tool for the comparison and quick access of large catalogues that is still not realistically feasible via the various VO tools due to the limitations of data transfer.

The choice of an underlying DBMS was, and still is, fundamental for the quick access to very large databases. There is the possibility to construct ad hoc programs that access flat files in a tree structure in a vary fast way as done originally in the release of the GSC2.3 or the USNOA, but these tend to be ad hoc or data specific and for them to be made general you will basically be repeating all the work done to develop a good DBMS without the software tools of a DBMS. The storing of the data under a DBMS has disadvantages: the user will have to install and manage the DB system on what ever computer they wish to use it; the data must be in some way dumped if you wish to exchange them with other users outside the DBMS; a DBMS has generally a larger overhead for small jobs; for simple catalogue access a good purpose built lower level program will probably be quicker in retrieving data for further manipulation. However, we believe these possible disadvantages are far outweighed by the benefits:

- the ability to treat widely varying datasets transparently with similar tools;
- the storage of maximum precision within the database ensuring that no precision is lost due to incorrect reading or writing of data;
- the saving of disk space due to storage within the database in binary format;
- the requirement on the user that tables and data within the database are sufficiently internally described and cataloged;
- the possibility to use DBMS systems that treat and search the data in very different ways depending on the problem being addressed;
- and, most importantly, the ability to on the fly index a given dataset in many different ways allowing a very quick access to parts of a large dataset via the index without the need to construct purpose built datasets or programs.

Finally, on a philosophical note, we believe that eventually most, if not all manipulation of scientific data should be done via a DBMS rather than as is currently the case using a variety of interim files because of the efficiency, ease and traceability of DBMS usage.

All information contained in this document is property of **INAF.** All rights reserved.

The DBMS we have chosen to use is MySQL because it was the best free well documented row based relational database on the market at the time. As we have said, if the problem changes and it would be better to have for example a column based relation database (e.g. Cache) or an object based database (e.g. objectivity) we can port into the new DBMS much of what we have developed, and in particular the various catalogues, with a minor investment of our time. Not to mention the fact that MySQL also offer “engines” that handle the data in a columnar way. An example is the InfiniDB engine (see http://www.infinidb.co/ and https://github.com/infinidb/). For now we are quite happy with the performance and power of MySQL and in comparisons to other free relational DBMSs (e.g. PostgreSQL) or commercial (e.g. Oracle) we do not find the difference in power or performance significant. We will briefly discuss here the TOCats database. Any subsets of this database, included CIWS Cat, are obviously subject to the same rules, procedures and access commands.

All information contained in this document is property of **INAF.** All rights reserved.

3. Accessing the data

Directly via the DBMS

To access the Mysql DBMS from a terminal on a computer with the TOCats (or its derivative) database installed, it is sufficient to type:

```
% mysql --database=TOCats --user=generic -password=password
```

or in short form:

```
% mysql TOCats -u generic -ppassword
```

or even shorter as this the default user: `% mysql TOCats`

Once in the Mysql you can run any mysql command which are fairly self explanatory:

```
mysql> show tables; mysql> describe GSC23;
mysql> select * from GSC23 limit 1;
mysql> help select;
```

The last command "`help command_name`" gives a very brief guide on how to use a given command. To find more detailed help it is better to use the online mysql help manual (http://dev.mysql.com/doc/#manual).

Via the WEB

A www catalogue server to extract objects from a circular or square region and to perform direct SQL queries is already available (http://parsec.oato.inaf.it/catalogues.php user: "generic" password: "password", or, http://ross.iasfbo.inaf.it/Ross/MyCatBrowser.php). This is just a preliminary version and gives access to just some catalogues, still it is usable by anyone. It is written in PHP with some Java-Script and Perl used for specific tasks. The web server is Apache and, again, the OS is Linux. So it uses the LAMP (Linux-Apache-MySQL-PHP) stack. The web based tools are by choice quite limited.

Via MCS

We expect people to carry out heavy access the database out via direct calls to the DB from lower level programming languages, e.g. C++, IDL, Fortran or PHP. This is achieved using the DB interfaces offered by the MCS library (http://ross.iasfbo.inaf.it/MCS/). To index these tables, which list celestial objects and therefore have spherical coordinates, and be able to quickly select the objects lying in a given sky region, we use the DIF package (see MCS site). It allows to convert a 2-d (index) spherical space into a 1-d space so that a standard B-tree index can be used. The package adds a series of functions and tools to the MySQL server and the table management is completely transparent to the user. The MCS software is open source freely available and while we use it here just as an interface to between the program and the DBMS, it is much more powerful allowing the user interactions with his/her machine. Indeed via MCS we can also perform queries to

All information contained in this document is property of **INAF.** All rights reserved.

databases on remote machines removing the requirement to have a working copy of the database on the local machine.

The procedures to access the database are very similar between the various programming languages and examples are provided at the MCS site for different languages.

All information contained in this document is property of **INAF.** All rights reserved.

4. TOCats

Each table in the database is equivalent to a catalogue, or in some cases of a very large catalogue, each table is part of the main catalogue. The TOCats database contains all the large optical and infrared catalogues publicly available: `GSC`, `USNO`, `PPMXL`, `2MASS`, `WISE`, `DENIS`, `SSA`, `SDSS`, etc. There are also many smaller astrometric or photometric catalogues: `UCAC`, `SPM/NPM`, `GSPC2`, etc. Finally there are many unpublished datasets such as parallax observations for the Torino Observatory Parallax Program (`PDAT_TOPP`), proper motions from the `PARSEC` ESO 2.2 parallax program (`PM_PARSEC`), proper motions from a combination of SDSS and GSC (`PM_GSCSDSS`), photometry from a TNG brown dwarf followup program (`PHOT_TNG`) or compilation catalogues such as the Initial Gaia Source List (`IGSL`). At the end of 2011 the number of tables was over 50 with numbers of entries ranging from a few 1000 to over 7 billion.

The descriptions of the tables and parameters are stored in the three tables `CatDesc`, `CatParams`, and `CatColDesc`. So to obtain basic source information on a given catalogue/table use the command:

```
select * from CatDesc where(TblName like "%GSC23%");
```

To obtain a list of columns for that table use the command:

```
select * from CatColDesc where(TblName like "%GSC23%");
```

These are descriptive tables provided to given the user immediately accessible information on the basic contents of the database and catalogues.

The table `CatParams` has a different scope and is being used together with the `DEFCatParams` table. They have stored all the metadata for each parameter in each table. The latter containing the information for all the commonly used parameter (column) names, like for coordinates, pixelization IDs, epoch, etc. This is primarily designed to be used by programs that wish to read in a semi automatic way our database. The C++ API developed within the CIWS does exactly that.

The command:

```
select distinct ColName from DEFCatParams;
```

lists the names of all those columns that are automatically managed for any catalogue and do not need to be specifically added into `CatParams`. For example to get the available metadat information for the RA coordinate (encoded in milli-arcsec):

All information contained in this document is property of **INAF.** All rights reserved.

```
select * from DEFCatParams where(ColName like "%RAmas%");
```

But for a column specific to a catalogue, like the R magnitude of the GSC23, we must use `CatParams`:

```
select * from CatParams where(TblName like "%GSC23%" and ColName like "%RGSC2mm%");
```

To know which columns specific to a catalogue are documented into CatParams, the query is simply:

```
select distinct ColName from CatParams where TblName="GSC23";
```

Example Parameters

Whenever possible we store parameters in the table in integer format. This is partially historical following the previous high cost of disk space but it also has the added advantage that the tables become very quick to search and index. For this reason the positions for example are converted to milli-arcseconds, the magnitudes are multiplied by 1000, and the epoch is multiplied by 100000 and all parameters sorted as integers. This also means the precision is inherent in the parameter's format and there is no lose of precision when transferring data between tables or reading into the various programming languages.

Here we list some of the most frequent parameters used in the database. As a rule of thumb the names of the parameters are of the format: `PARAMunit` and the associated errors `ePARAMunit`.

Equatorial positions are given as `RAmas, DECmas` with errors (when present) `eRAmas, eDECmas` always in the system ICRF J2000 and units milli-arcseconds. Proper motions (`RAPMdmas, DECPMdmas`) and related errors (`eRAPMdmas, eDECPMdmas`) are given as 10*mas/yr to allow these values to not lose precision when converted to integer values.

Magnitudes are provided with names that include the natural filter system, hence we have `RGSC2mm, RSDSSmm` and `Rmm` indicating a R magnitude in the natural system of the `GSC2, SDSS` and Cousins standard system, similarly for B, V, and I. For IR magnitudes the example would be `J2MASSmm, JDENISmm and JMKOmm` and similarly for H and K. There exists many other magnitudes in the various tables as they are often not standard and the only way to have general programs is to rely on the metadata in the `CatParams` table.

All information contained in this document is property of **INAF.** All rights reserved.

Other important variables are the HTM or HEALPix pixel IDs `htmID_6` or `healpID_nest_8` which are the pixelisation of the sky used by the various cone search routines in DIF. `CLASSIFICATION` which provides a classification of each object as provided by the various sky surveys. Various flags such as `GSC23.sourceStat, TMASS.FLAGStmassg, UCAC.stat1/stat2`, etc. The epoch is coded into a variable `EP00000c` which is the (quoted epoch of position – 1900)*100000. All of these and other variables can be decoded using the metadata.

Finally there is the `MASTERhpx6` and `runningnumber` parameters. This is a "name" given to an object if it is matched in a master list call `MASTER` made up of a compilation of the large infrared and optical surveys. `MASTERpx6` is the HEALPix level 6 value for the first inputted position and `runningnumber` is the object number within that pixel. This is the basis for the Gaia `sourceID` and is used to allow a quick cross correlation of the different catalogues. The matching is done using a large radius (5") and based on a nearest neighbor with no possibility of having more than one match in any given catalogue, unless explicitly allowed such as the case of multiple observations of a given object. This is robust and for statistical purposes more than sufficient, if however more refined matching is needed the user is encouraged to develop their own routines that are tailored to their purposes.

SQL Facilities

Users can use or build custom queries/tools to access the metadata info. However we have developed some SQL functions and procedures that can both be used directly or taken as examples to implement more of them. In particular we have developed a number of functions that return information if the form of JSON strings (see json.org). This is very useful to easily manage the info in user's applications. A preliminary list follows:

`encoords.sql`:

It contains the functions `RAstd` and `DECstd` which convert input RA and DEC from degrees to standard format "hh mm ss.ss" and "±dd mm ss.s", respectively.

`cat_field_names.sql`:

Function to retrieve a simple comma separated list of column names for a given catalogue and (optional DB name - see header comment).

`cat_mag_names.sql`:

Function to retrieve a simple comma separated list of column names for the magnitudes in a given catalogue in the form:

"`ColMag1*1e-3 as R, ColMag2*1e-3 as B` ...".

`cat_field_offset.sql, cat_field_scale.sql`:

All information contained in this document is property of **INAF.** All rights reserved.

Functions to retrieve numeric `OFFSET`, `SCALE` parameters for a given column and catalogue.

`jcat_field_types.sql`, `jcat_field_params.sql`, `jcat_info.sql`, `jcat_field_scaleoffs.sql`, `jcat_field_outnames.sql`, `jcat_field_outunits.sql`:

Functions to retrieve JSON formatted info from `CatParams` (see header comments).

`cat_stdval_sqry.sql`:

Function to build a prepared SELECT query to get output values in standard format (use `OFFSET`, `SCALE` and `OUTNAME`).

`catcircle_std_values.sql`, `catsquare_std_values.sql`:

Procedures to list all the fields of a catalogues in standard units for a given circular or square sky region. They assume to have HTM DIF indexed catalogues.

All information contained in this document is property of **INAF.** All rights reserved.

5. CIWS Cat

The catalogue produced for the Customizable Instrument Workstation Software system is a combination of the following TOCats catalogues:

GSC2.3 – *The Second Guide Star Catalog version 2.3 Lasker et al (2008, AJ, 136, 735).*

This catalogue forms the bulk of the photometry and defines the red and blue magnitudes as this is the sky survey with the largest coverage on a precise homogenous photometric system. The only variation with the public version is that we removed the multiple entries discussed in section 4.2 of Lasker et al (2008). This was done by insisting that only the Tycho-2 or first occurrence of entries within 10 mas were kept. As described in section 4.3 of Lasker et al (2008) the errors are not formal errors and cannot be used to set confidence limits hence we have set the position errors to 400 mas.

2MASS – *Two Micron All-Sky Survey Point Source Catalog Epchtein et al., (1999, A&A, 349, 236).*

This catalogue is used to provide Infrared magnitudes J, H, K. As we only considered the PSC all objects in this catalogue were classified as stellar.

WISE – *Wide-field Infrared Survey Explorer Wright et al. (2010,AJ 140.1868).*

It provides the long 3-7 micron magnitudes in bands W1, W2, W3 and W4.

The parameters in the `CIWScat` MySQL table are positions (`RAmas, DECmas` in milliarcseconds), proper motions (`RAPMdmas, DECPMdmas` in 10*mas/yr), optical GSC23 blue, visual, red and far-red magnitudes (`BGSC2mm, VGSC2mm, RGSC2mm, IGSC2mm` in millimags), infrared 2MASS J, H, K magnitudes (`J2MASSmm, H2MASSmm, K2MASSmm` in millimags) and the far infrared WISE magnitudes (`W1mm, W2mm, W3mm, W4mm`, along with a classification and ID (`CLASSIFICATION, MASTERhpx6 + runningnumber`). With the portable software we have provided a default version, `CIWScat14`, that contains **20,206,064** entries and a cut in GSC23 red magnitude of 14. The catalogue size is about 1 GB and contains 20,206,064 objects. The software can also remotely access the `CIWScat18` which has a magnitude cut of 18 and has **326,999,940** entries. This catalogue is about 21 GB in size.

As mentioned above, the stand-alone program `getCIWScat` offers various query options and output formats to the user, and they can easily be extended if necessary. To see the available options it's enough to give the command without parameters:

```
% GetCIWScat
getCIWScat  Ver 0.1b, 05-Mar-2014, LN@IASF-INAF


Usage:
  getCIWScat [OPTIONS] RAcenter DECcenter side_arcmin [CatName]
or:
```

All information contained in this document is property of **INAF.** All rights reserved.

```
  getCIWScat [OPTIONS] RAcorner1 DECcorner1 RAcorner2 DECcorner2 [CatName]
with coordinates in degrees, or:
  getCIWScat [OPTIONS]  hh mm ss  +/-dd mm ss  side_arcmin  [CatName]
                       (RAcenter) (DECcenter)
Where OPTIONS are:
  -hcCeEinqrS [-d DB_name] [-m Mag_lim] [-o Order_By] [-O Order_By] [-s
Server_name]
  -h: print this help
  -c: compact version: just print 'RA (deg) Dec (deg) R  H'
  -C: input data are center and radius of a circle
  -e: extended version: print more all the info read
  -E: use external server (Turin/Bologna)
  -i: as -q but does not check existence of CatName
  -n: just print the number of objects resulting from the query
  -q: print the query without executing it
  -r: print the output in raw format as read from the DB
  -S: print RA and Dec in string rather than fractional degrees
  -d DB_name: send query to 'DB_name'
  -m Mag_lim: select only objects with (one or more) Mag < 'Mag_lim'
  -o Order_By: order output using 'Order_By' column (use -r to see names)
  -O Order_By: like -o but sort in Descending order
  -s Server_name: send query to 'Server_name'
and 'CatName' can be:
  CIWScat14 (def.) | CIWScat18 | (more to come) |
':' and fractional 'ss.s' in coordinates are allowed.
Default server and DB: localhost, PUBCats
```

Here are some examples of usage:

```
% getCIWScat 10 20 30
#RA Dec B V R I J H K W1 W2 W3 W4
10.237224 +19.892600 14.61 13.99 13.66 13.34 12.90 12.59 12.52 -32.77 -32.77
-32.77 -32.77
10.145366 +19.989884 14.89 14.23 13.94 13.70 13.08 12.80 12.71 -32.77 -32.77
-32.77 -32.77
...
```

`% getCIWScat 10 12 13 -20 30 40 10` (or `getCIWScat 10:12:13 -20:30:40 10`)

All information contained in this document is property of **INAF.** All rights reserved.

```
#RA Dec B V R I J H K W1 W2 W3 W4
153.106171 -20.501918 14.34 -32.77 12.81 12.21 11.34 10.74 10.63 -32.77 -32.77
-32.77 -32.77
153.088559 -20.464719 14.23 -32.77 13.27 13.02 12.57 12.25 12.23 -32.77 -32.77
-32.77 -32.77
...

% getCIWScat -c 10 -20 30

#RA Dec R H
10.039079 -20.048558 12.49 11.62
10.068512 -20.029344 13.93 12.54

% getCIWScat -c -S 10 -20 30

#RA Dec R H
10:02:20.68 -20:02:54.8 12.49 11.62
10:04:06.64 -20:01:45.6 13.93 12.54
...
```

All information contained in this document is property of **INAF.** All rights reserved.