CALIFA table format specification

Document created by the DR working group

Released on the 13th of September 2011
Last updated on the 20th of October 2011

1 Format specification for CALIFA tables

1.1 Aims

For the exchange of important information on the characterization of the CALIFA sample as well as to create a dedicated database system for the public CALFIA data release a standard for the format of tables need to specified. The two common table formats widely used in astronomy are ASCII and FITS tables. Here we define some rules and standards for both table formats the collaboration must follow to ensure a minimum level of homogeneity, prevent loss of information and allow an easy ingest to a database.

To create a table for CALIFA either the FITS or ASCII format may be used. It will automatically be converted to the other format by the Data Release working group. Together with the table an associated ASCII file needs to be created that describes in detailed how the content of the table was generated. Until a dedicated data management system is running for CALIFA both tables formats will be posted on the WIKI.

1.2 ASCII tables

ASCII tables must consist of a header providing required meta data followed by the data section with the actual content of the table.

1.2.1 Header section

The header of the ASCII files must have a form similar to the following example:

```
# AUTHOR: Carlos C. Califa
# SOURCE: CALIFA Collaboration
# DATE: 2011-08-24
# VERSION: 1.0
# COLAPRV: J. Walcher
# PUBAPRV: None
# COLUMN1: CALIFAID, int, , the ID of the CALIFA galaxy
# COLUMN2: CALIFAName, string, , the CALIFA name of the galaxy
# COLUMN3: Name, string, , the NED name of the galaxy
# COLUMN4: RA, float, degrees, right ascension J2000.0
# COLUMN5: DEC, float, degrees, declination J2000.0
```

The different keywords are:

- `AUTHOR`: creator of this particular file
- `DATE`: date on which the table was created (the format is `yyyy-mm-dd`)

- `VERSION`: version of this particular table
- `SOURCE`: sources of the data, e.g. a survey, a collaboration, a paper
- `COLAPRV`: Name of the person who approved the table for internal release, (the default is "None" and will be later changed by the person in charge)
- `PUBAPRV`: Name of the person who approved the table for public release, (the default is "None" and will be later changed by the person in charge)
- `COLUMNn`: description of the column `n`. It must contain 4 fields:
  - Column name: This will be the column name in the database, therefore, it should be short and concise, and it must contain only letters, numbers, and the underscore (in particular no whitespaces).
  - Data type: The possible types for the column are: string, short, int,long, float, double (see below for further specification of the possible types)
  - Physical units: The physical unit in which the column's values are given, e.g. deg, Mpc, etc. (if no unit applies this field is left blank).
  - Brief description: This field should contain all information, which is needed for the usage of the column's values. Note, that this information should not be in the column's name, making it as concise as possible.

1.2.2 Data section

The header is followed by the data section were the different values for each column are separated by commas in a row. Based on the header example above the data section would look like this:

```
1, CALIFA001, IC5376, 0.33241081, 34.52566909
2, CALIFA002, UGC00005, 0.77351248, -1.91383457
3, CALIFA003, NGC7819, 1.10210525, 31.47200775
4, CALIFA004, UGC00029, 1.14060366, 28.30172348
5, CALIFA005, IC1528, 1.27240324,-7.09338998
...
```

1.3 Binary FITS tables

1.3.1 Table FITS header

The FITS header of the table must contain the following keywords similar to those specified for the ASCII header:

- `AUTHOR`: creator of this particular file
- `DATE`: date on which this particular file was created (the format is `yyyy-mm-dd`)
- `VERSION`: version of this particular table
- `SOURCE`: sources of the data, e.g. a survey, a collaboration, a paper

- `COLAPRV`: Name of the person who approved the table for internal release, (the default is ”None” and will be later changed by the person in charge)
- `PUBAPRV`: Name of the person who approved the table for public release, (the default is ”None” and will be later changed by the person in charge)
- and for each column (`n` represent the number of a particular column):
  - `TTYPEn`: Column name: This will be the column name in the database, therefore, it should be short and concise, and it must contain only letters, numbers, and the underscore (in particular no whitespaces).
  - `TFORMn`: Data type: This field is usually automaticly created, the possible values for CALIFA DR1 are: I,J,K,E,D, or ?A, where ? denotes an integer. (see below for further specification of the possible types)
  - `TUNITn`: Physical units: The physical unit in which the column's values are given, e.g. deg, Mpc, etc. (if no unit applies this field can be ommited).
  - `TCOMMn`: Brief description: This field should contain all information, which is needed for the usage of the column's values. Note, that this information should not be in the column's name, making it as concise as possible.
  - `TNULLn`: NULL value: This field contains the NULL value for the the for column (only for integer types). For further information see below.

1.4 Reference column

Each table must contain a reference column with a unique identifier in each row. For tables describing primary galaxy properties like morphology or magnitudes for each CALIFA galaxy, the reference column must be the CALIFAID column as the first column in a table. The reference column should be the first column of the table.

1.5 Data types

We support six different data types for the columns. In ASCII files they must be specified in the header using the C/C++/Java like identifiers given in the table below. In FITS files the usual formats in the header apply.

ASCII	FITS	Explanation
short	I	16-bit signed integer
int	J	32-bit signed integer
long	K	64-bit signed integer
float	E	32-bit single precision floating point
double	D	64-bit double precision floating point
string	?A	String in ASCII format/Number of characters for FITS

1.5.1 Units

In principle any system of units is accepted for a given column, as long as the values can be expressed by integers or floating point numbers. Right ascension and declination and any other angle must be given in degrees. Hours, minutes, and seconds will not be supported. The coordinates should be provided for the J2000 epoch. In any case indicate the epoch of the coordinates in the description field, so that it can always be converted, if required.

1.5.2 NULL values

If a particular value cannot be provided, a special marker must be used to indicate a `NULL` value:

- for ASCII files: the value is omitted, i.e. two commas appear next to each other
- for FITS files: the used marker depends on the data type of the column:
  - for strings: an empty string is used
  - for float and double: the `NaN` value is used
  - for short, integer and long: a user-give value is used, which must be specified using the `TNULLn` keyword in the FITS header. Of course the chosen value needs to be far outside the range of the other values for this column.

1.5.3 Special characters

In ASCII format, strings must not contain commas or colons, unless the whole string is quoted by double quotes. For column names no special character besides the underscore is allowed.

1.6 Associated ASCII file

Together with the table an associated ASCII file needs to be created which describes more detailed the creation and origin of the table. Since this will very much depend on the information content of the table, no particular format is defined except that it should start with the three keywords `# AUTHOR:`, `# Date:`, `# Version:` like in the ASCII table file.

The rule should be that it contain as much information to independently reproduce the content of the table, i.e. state the SQL Query to extract information from an external database, source of data and analysis steps that yield the provided data. Each time a new version of the table is generated this file must to be extended to describe any changes regarding the previous version.

1.7 File naming

The name of the two table formats and the associated ASCII file need to have an identical file name to ensure a unique association except of the extension:

- `.csv` for ASCII tables
- `.fits` for FITS tables
- `.txt` for the associated files

1.8 Ancillary data table

Ancillary data sets should be stored on the CALIFA FTP Server in the sub-directory `ftp://ftp.caha.es/ancillary/`. For each set (which is allowed to contain several files per object) a seperate sub-directory should be used. In addition to these files, a corresponding table must be created to map the CALIFAID to the corresponding ancillary data files (as the relative path from the ancillary directory). Like for regular tables, associated ASCII file that describes the origin of the data must also be provided. The table and the associated ASCII file must be placed in the ancillary directory and should have the same (meaningful) name as the sub-directory containing the ancillary data set.

1.9 Updating tables

Certainly, the content of tables will change over time which need to be updated in the database. There will be two kinds of changes, minor and major updates. In the first case either the table contained some small bugs or the table format did not comply with our defined standards. These will be fixed by created a new table with the same version number that replaces the corresponding table in the database. In the second case a major part of the table content has changed, e.g. columns were added or the content of column(s) changed. A completely new table needs to be created with a new version number in this situation that will be *added* to the database.

2 Examples

2.1 ASCII file

```
# AUTHOR: C.J. Walcher
# SOURCE: CALIFA Collaboration
# DATE: 2011-09-30
# VERSION: 1.0
# COLAPRV: J. Walcher
# PUBAPRV: None
# COLUMN1: CALIFAID, int, , the ID of the CALIFA galaxy
# COLUMN2: RA, float, degrees, right ascension J2000.0
# COLUMN3: DE, float, degrees, declination J2000.0
# COLUMN4: hubtyp, string, , Hubble type S or E
# COLUMN5: hubsubtyp, string, , Hubble subtype 0-7 or a-d
# COLUMN6: minhubtyp, string, , Minimum Hubble type
# COLUMN7: minhubsubtyp, string, , Minimum Hubble subtype
# COLUMN8: maxhubtyp, string, , Maximum Hubble type
# COLUMN9: maxhubsubtyp, string, , Maximum Hubble subtype
1, 0.33241081, 34.52566909, S, b, S, a, S, d
2, 0.77351248, -1.91383457, S, bc, S, ab, S, c
3, 0.10210525, 31.47200775, S, c, S, bc, S, c
4, 1.14060366, 28.30172348, E, 1, E, 0, E, 1
5, 1.27240324, -7.09338998, S, bc, S, ab, S, d
...
```

2.2 FITS file (header only)

```
AUTHOR  = ’C.J. Walcher’
SOURCE  = ’CALIFA Collaboration’
DATE    = ’2011-09-30’
VERSION = ’1.0     ’
COLAPRV = ’J. Walcher’
PUBAPRV = ’None    ’
TTYPE1  = ’califaid’
TFORM1  = ’J       ’
TCOMM1  = ’the id of the califa galaxy’
TNULL1  =           2147483647
TTYPE2  = ’ra      ’
TFORM2 = ’E       ’
TUNIT2  = ’degrees ’
TCOMM2  = ’right ascension J2000.0’
TTYPE3  = ’de      ’
TFORM3  = ’E       ’
TUNIT3  = ’degrees ’
TCOMM3  = ’declination J2000.0’
TTYPE4  = ’hubtyp  ’
TFORM4  = ’128A    ’
TCOMM4  = ’hubble type s or e’
TTYPE5  = ’hubsubtyp’
TFORM5 = ’128A    ’
TCOMM5  = ’hubble subtype 0-7 or a-d’
TTYPE6  = ’minhubtyp’
TFORM6  = ’128A    ’
TCOMM6  = ’minimum hubble type’
TTYPE7  = ’minhubsubtyp’
TFORM7  = ’128A    ’
TCOMM7  = ’minimum hubble subtype’
TTYPE8 = ’maxhubtyp’
TFORM8 = ’128A    ’
TCOMM8 = ’maximum hubble type’
TTYPE9 = ’maxhubsubtyp’
TFORM9 = ’128A    ’
TCOMM9 = ’maximum hubble subtype’
```

2.3 Associated ASCII file for table

```
# AUTHOR: C.J. Walcher
# SOURCE: CALIFA Collaboration
# DATE: 2011-09-30
# VERSION: 1.0

This catalogue is based on eyeball classification of the galaxies in the CALIFA
```

mothersample by Raffaela Ana-Marino, Jairo Mendez-Abreu, Vesselina Kalinova, Nadine Backsmann and Clara Cortijo. They classified the galaxies according to:

The columns should be:
1) E or S or I for elliptical, Spiral, Irregular
2) 0-7 (for Es) or 0, 0a, a, ab, b, bc, c, cd, d, m (for S) or r (for I)
3) B for barred, otherwise A. AB if unsure.
4) Merger features (M) or Isolated (I)
For mergers fill columns 1 to 3 with the properties of the main object,
if possible. If nothing at all is possible you may write U (unknown) there.

The postage stamps on the CAHA ftp (r and i band) were used.

What I have done then is very simple:

1) Use all catalogues.
2) I used the translation given below to turn class into numbers.
After the translation every galaxy has three numbers attached,
one for morphology, one for barredness and one for interaction state.

```
IF (class[0] EQ 'E') THEN begin
IF (class[1] EQ '0') THEN numb[0] = 0
IF (class[1] EQ '1') THEN numb[0] = 1
IF (class[1] EQ '2') THEN numb[0] = 2
IF (class[1] EQ '3') THEN numb[0] = 3
IF (class[1] EQ '4') THEN numb[0] = 4
IF (class[1] EQ '5') THEN numb[0] = 5
IF (class[1] EQ '6') THEN numb[0] = 6
IF (class[1] EQ '7') THEN numb[0] = 7
ENDIF
IF (class[0] EQ 'S') THEN begin
IF (class[1] EQ '0')  THEN numb[0] = 8
IF (class[1] EQ '0a') THEN numb[0] = 9
IF (class[1] EQ 'a')  THEN numb[0] = 10
IF (class[1] EQ 'ab') THEN numb[0] = 11
IF (class[1] EQ 'b')  THEN numb[0] = 12
IF (class[1] EQ 'bc') THEN numb[0] = 13
IF (class[1] EQ 'c')  THEN numb[0] = 14
IF (class[1] EQ 'cd') THEN numb[0] = 15
IF (class[1] EQ 'd')  THEN numb[0] = 16
IF (class[1] EQ 'dm') THEN numb[0] = 17
IF (class[1] EQ 'm')  THEN numb[0] = 18
ENDIF
IF (class[0] EQ 'I') THEN numb[0] = 19

IF (class[2] EQ 'A') THEN numb[1] = 0
IF (class[2] EQ 'AB') THEN numb[1] = 1
IF (class[2] EQ 'B') THEN numb[1] = 2

IF (class[3] EQ 'I') THEN numb[2] = 0
IF (class[3] EQ 'M') THEN numb[2] = 1

3) I calculated the mean of these three number for each galaxy
4) All classifications more than 4 away from the mean were excluded
and mean recalculated.
5) The minimum and maximum number among ALL classifications is also
kept and serves as upper and lower limit.
6) Numbers are translated back into morphological code and written to table.
```

2.4 A dummy ASCII file for ancillary data files

```
# AUTHOR: Carlos C. Califa
# SOURCE: CALIFA Collaboration
# DATE: 2011-08-25
# VERSION: 1.0
# COLAPRV: J. Walcher
# PUBAPRV: None
# COLUMN1: CALIFAID, int, , the ID of the CALIFA galaxy
# COLUMN2: sdss_u_img, string, , relative path to sdss u band image
# COLUMN3: sdss_g_img, string, , relative path to sdss g band image
# COLUMN4: sdss_r_img, string, , relative path to sdss r band image

1, sdss_3x3/CALIFA001_u.fits, sdss_3x3/CALIFA001_g.fits, sdss_3x3/CALIFA001_r.fits
2, sdss_3x3/CALIFA002_u.fits, sdss_3x3/CALIFA002_g.fits, sdss_3x3/CALIFA002_r.fits
3, sdss_3x3/CALIFA003_u.fits, sdss_3x3/CALIFA003_g.fits, sdss_3x3/CALIFA003_r.fits
4, sdss_3x3/CALIFA004_u.fits, sdss_3x3/CALIFA004_g.fits, sdss_3x3/CALIFA004_r.fits
5, sdss_3x3/CALIFA005_u.fits, sdss_3x3/CALIFA005_g.fits, sdss_3x3/CALIFA005_r.fits
...
```

3 Important things to consider

- Do not mix up the meaning of column name and column description! Please use *concise* column names (less than 10 characters as a guideline), which must contain only letters, numbers, and the underscore, but particularly *no whitespaces*

- Do not forget to provide a column description for each column as well as a physical unit. If no physical unit can be assigned to a specific column use a white space instead in the ASCII table format or leave it out in the FITS table format

- A reference column with a unique identifier must be present, i.e. the CALIFAID in most cases

- Prepare an associated ASCII file for your table that describe how the content of the table was generated

- Choose the proper file extension .csv, .fits, .txt for the files with the same file name
- ASCII table format: strings must not contain commas or colons, unless the whole string is quoted by double quotes
- ASCII table format: comma should also be used as the separator for the data section of the table
- ASCII table format: check that all columns in the data section are properly defined in the header section and that their numbering is correct
- FITS table format: take care about NULL values