Data Central Ingestion

Release 1.10

Liz Mannering

Sep 20, 2023

CONTENTS

Data Central simplifies astronomical data releases by automating the ingestion, versioning, web-based data access, and hosting of your data and supporting documentation.

Data Central has many features for hosting astronomical data including:

- Image cutout service produces publication-quality single-band or rgb images with custom image scaling options, fits creation, overplotting sources from DC-hosted and remote catalogues, batch mode, contours and more, over 30 bands across the GAMA and DEVILS regions (with VHS data coming soon)
- Single object viewer including image cutout at source position, data product download, 2D map visualizer, tabular data
- Cone search service including remote source position overlays
- Auto-generated, interactive, intuitive schema browser
- SQL/ADQL query service; return records that match some criteria from the catalogues hosted at Data Central
- TAP service
- Authenticated users can re-visit their query/cutout/cone search history

Your data will be discoverable alongside many of Australia's national optical surveys. Data Central also provides access to our in-house CMS, Document Central, for your survey team to curate the data-release's supporting documentation.

Tip: Data Central hosts a few different types of documentation:

- static documentation on a specific data release (appearing in the schema browser)
- updatable documentation about a survey (Document Central)
- team wikis
- documentation on how to use Data Central itself (including how to cite Data Central and the surveys we host).

Only the first of these is discussed in the ingestion documentation, and only the first of these needs to be provided by the survey/data team before ingestion can begin. The others can be changed at any time, outside of the constraints of the ingestion process.

Think of the first as a paper—you cannot change it once you've published it, you can only publish a new paper with the corrections (allowing citations)—whereas the others are like wikipedia, in that the content can be continuously updated, and hence is a poor source for citation.

CHAPTER

ONE

GETTING STARTED

Data Central can currently ingest, link, cross-match and serve spectra, images, catalogues and IFS data products. We have detailed descriptions for how to format your data depending on its type, but first, follow the instructions below to setup the basic file structure and metadata files for your data release.

Metadata files are provided alongside your data and allow us to populate the Schema Browser to best describe your data. These files are un-editable once your data has been released via Data Central (prior to the final sign-off from the survey team, we will go through an iterative process with you to verify that the data are correct and displayed as expected. During that stage you are free to make changes to metadata files).

Documentation intended for public consumption that you will want to update frequently (e.g., with detailed descriptions of analysis or version update information) should be maintained by the survey teams themselves in Data Central's Documentation portal: Documentation Central. Contact us to get your team set up within the system if you have not done so already.

Whilst we can be flexible about how we receive the data and metadata, here's what we see as the quickest and easiest way to get your data ingested:

1. Create a Data Central Account at https://accounts.datacentral.org.au/register/ (if you have not already done so).
2. Request a team at https://teams.datacentral.org.au/request-team/ (if you do not already have a team for the survey/data). The team will need to be approved by a Data Central admin.
3. Log in with your Data Central account at https://dev.aao.org.au/ (using the Data Central button), as this will allow you git access to the repository which stores all the metadata. If multiple members of the team want to be updating the metadata, have them also create Data Central accounts and login to https://dev.aao.org.au, and include their usernames in a ticket here https://jira.aao.org.au/servicedesk/customer/portal/3/create/28.
4. Upload your data in the correct layout (as noted below) to https://cloud.datacentral.org.au/.
5. Make a merge request on https://dev.aao.org.au/ with the metadata as documented below.
6. Check the data in the pre-release link we (the Data Central team) send you, and sign off on the release.

1.1 Directory Structure

To ingest data into Data Central, you will provide two folders, one containing the data products themselves, and one containing the metadata. The data/ and metadata/ directories will be further populated according to the types of data products you wish to release (see section *Data Types* for specific requirements; catalogue/ifs/spectra etc).

1.1.1 Data

A top-level `<survey>` directory should contain a single directory per `<datarelease>`.

```
data
└── <survey>
    └── <datarelease>
```

1.1.2 Metadata

The following file structure should be adopted. A top-level <survey> directory should contain a single directory per <datarelease>. Both directories should have metadata files described below which will populate the Schema Browser.

```
metadata
└── <survey>
    ├── <survey>_survey_meta.txt
    └── <datarelease>
        └── <survey>_<data_release>_data_release_meta.txt
```

1.2 Metadata Files

Attention: Metadata files are always pipe-delimited, and have the extension .txt

There are two files you'll need to provide to initialize your survey and data release in Data Central, *<survey>_survey_meta.txt* and *<survey>_<datarelease>_data_release_meta.txt*.

1.2.1 <survey>_survey_meta.txt

This file describes your survey at a high level, and is used to populate the Schema Browser. It is uneditable once the data are released, detailed documentation that is editable by the teams should be written into a Document Central article (contact us if you do not yet have an account).

Provide the following a single pipe-delimited .txt file containing an entry (row) for your survey:

name	pretty_name	title	description	pi	contact	website
gama	GAMA	Galaxy and Mass Assembly Survey	GAMA is a project to exploit the latest generation of ground-based and space-borne survey facilities to study cosmology and galaxy formation and evolution. At the heart of this project lies the GAMA spectroscopic survey of ~300,000 galaxies down to r < 19.8 mag over ~286 deg2, carried out using the AAOmega multi-object spectrograph on the Anglo-Australian Telescope (AAT) by the GAMA team. This project was awarded 210 nights over 7 years (2008–2014) and the observations are now completed. This survey builds on, and is augmented by, previous spectroscopic surveys such as the Sloan Digital Sky Survey (SDSS), the 2dF Galaxy Redshift Survey(2dFGRS) and the Millennium Galaxy Catalogue (MGC).	Simon Driver	Simon Driver	http://www/gama-survey.org/

Replace <survey> in the filename with the value for *name* in the first column.

<survey>_survey_meta.txt

This file should contain the following columns

name (*required=True*, *type=char*, *max_limit=100*)
A machine-readable, lowercase version of your survey name (e.g., GAMA would be entered as gama). Use only alphanumeric characters (no spaces and do not start your survey with a digit). This must be unique within Data Central (we cannot support multiple surveys named 'DEVILS', you will need to think of a new name!). This should match the string used in the file name.

pretty_name (*required=True*, *type=char*, *max_limit=100*)
A human-readable version of the survey name. This can contain any characters (up to the character limit).

title (*required=True*, *type=char*, *max_limit=100*)
A longer version of the survey name (likely an expansion of the acronym!)

description (*required=True*, *type=char*, *max_limit=1000*)
A succinct paragraph describing your survey.

pi (*required=True*, *type=char*, *max_limit=100*)
The name of the Principal Investigator

contact (*required=True*, *type=char*, *max_limit=100*)
Format as: John Smith <john.smith@institute.org>

website (*required=True*, *type=char*, *max_limit=500*)
The survey team's website for public consumption (e.g., https://devilsurvey.org/)

Note: If you have previously hosted data with Data Central, you can skip providing the <survey>_survey_meta.txt file. We'll already have this from your previous data release.

1.2.2 <survey>_<datarelease>_data_release_meta.txt

This file describes your data release at a high level, and is used to populate the Schema Browser. It is uneditable once the data are released, detailed documentation that is editable by the teams should be written into a Document Central article (contact us if you do not yet have an account).

Provide the following a single pipe-delimited .txt file containing an entry (row) for your data release:

name	pretty_name	version	data_release_number	contact
dr2	Data Release 2	1	2	Simon Driver <simon.driver@uwa.edu.au>

Replace `<datarelease>` in the filename with the value for *name* in the first column.

<survey>_<datarelease>_data_release_meta.txt
This file should contain the following columns

name (*required=True*, *type=char*, *max_limit=100*)
A machine-readable, lowercase version of your data release name (e.g., Data Release 2 would be entered as dr2). Use only alphanumeric characters (no spaces). This must be unique within your survey (you cannot release dr2 twice!). This should match the string used for the data release in the file name.

pretty_name (*required=True*, *type=char*, *max_limit=100*)
A human-readable version of the data release name. This can contain any characters (up to the character limit).

version (*required=True*, *type=float*, *default=1.0*)
The version of the data release data. This is meaningful to the team only, and should be described in the team-curated and created Documentation

`data_release_number` (*required=True*, *type=int*, *default=1*)
: The number of the data release within Data Central. e.g., if this data release is named "Final Data Release", which may correspond to the 5th data release, we need a numeric representation here so as to show the data releases in the schema browser sequentially.

`contact` (*required=True*, *type=char*, *max_limit=100*)
: Format as: John Smith <john.smith@institute.org>

The basic models are now also in place to begin ingesting data. Since your data release will likely contain catalogues, we recommend you start there first.

Read Next:

Catalogues

Spectra

IFS

Images

CHAPTER

TWO

DATA TYPES

Currently Data Central supports catalogues, IFS, spectra and imaging data. View the sections below for details on file formatting and the required metadata for each data type.

Warning: Please be sure to read the documentation carefully. Pay attention to restrictions on the contents of columns (e.g. restictions on allowed characters, choosing between a limited number of options), and double check that names match across different files. This reduces the number of issues encountered in ingestion, and reduces the number of iterations needed to finalise the data release.

2.1 Catalogues

Ingesting catalogues into Data Central allows for SQL/ADQL querying, and broadcasts the table(s) through the Data Central TAP server. The Schema Browser will also include an entry for each catalogue.

If you provide an input catalogue as part of the catalogue ingestion (as described later in this article), additional functionality is provided:

- sources will appear in the Name Resolver and can be automatically resolved by the image cutout.
- sources will appear in the Cone Search
- sources will be available in the Single Object Viewer. Individual data products (IFS, Spectra) can be linked to a source, and custom SQL run to populate the SOV with particular rows from your catalogues.

Note: `source` in Data Central is used interchangeably with `AstroObject`. It is a survey-team defined astronomical object with positional information that individual data product files can be linked to.

Note: Remember that the documentation mentioned here is the static, paper-like documentation, the documentation on Documentent Central is entirely separate.

2.1.1 Data Model

Data Central's ingestion process will map your data onto the Data Central data model format. Within Data Central, catalogue data are organised hierarchically, as per:

```
Survey
└── DataRelease
    └── Schema:Catalogues
        └── Group
            └── Table
```

There are dozens of tables from multiple surveys in the Data Central database. `Groups` are used to collect scientifically-related tables together, in order to help the user locate the correct table more quickly. To explore the data model further, visit the catalogue section of the Schema Browser to explore the relationships between groups and tables.

2.1.2 Directory Structure

To ingest data into Data Central, you will provide two folders, one containing the data products themselves, and one containing the metadata.

Data

The catalogues directory should contain the catalogue files themselves.

```
data
└── <survey>
    └── <datarelease>
        └── catalogues
            ├── my_input_cat.fits
            └── my_output_table.csv
```

Attention: `<survey>` and `<datarelease>` should be replaced with the values you chose in *Getting Started*, e.g., gama and dr2

Data Central supports catalogues/tables in .csv or .fits formats.

Danger: If your input table is > 2GB in size, please ensure the format is .csv (not fits).

Metadata

The following file structure should be adopted. A top-level `<survey>` directory should contain a single directory per `<datarelease>`. Both directories should have metadata files described below which will populate the Schema Browser.

```
metadata
└── <survey>
    ├── <survey>_survey_meta.txt
    └── <datarelease>
        ├── <survey>_<data_release>_data_release_meta.txt
        └── catalogues/
            ├── <survey>_<datarelease>_column_meta.txt
            ├── <survey>_<datarelease>_coordinate_meta.txt  ** optional
            ├── <survey>_<datarelease>_group_meta.txt
```

(continues on next page)

(continued from previous page)

```
              ├── <survey>_<datarelease>_sql_meta.txt         ** optional
              ├── <survey>_<datarelease>_table_meta.txt
              └── docs/
```

The metadata/catalogues/ directory will contain a minimum of 3 metadata files, plus a docs/ directory if you have supplied additional documentation for a particular catalogue. The two optional metadata files (coordinate_meta and sql_meta) are described later in this article.

2.1.3 Metadata Files

Attention: Metadata files are always pipe-delimited, and have the extension .txt

<survey>_<datarelease>_group_meta.txt

This file describes the `groups` you would like to register, and will be used to populate the Schema Browser. Provide the following a single pipe-delimited .txt file containing an entry (row) for each group:

name	pretty_name	description	documentation	contact	date	version
ApMatchedPhotom	ApMatchedPhotom	This group provides aperture matched ugrizYJHK photometry.	unique_group_documentation_filename.txt	name <email@institute.org>	2012-04-23	v02

Please name this file: <survey>_<datarelease>_group_meta.txt e.g., sami_dr2_group_meta.txt

`<survey>_<datarelease>_group_meta.txt`

This file should contain the following columns

`name` (*required=True*, *type=char*, *max_limit=100*)

Group name. Use only alphanumeric characters. This must be unique per data release.

`pretty_name` (*required=True*, *type=char*, *max_limit=100*)

A human-readable version of the group name. This can contain any characters (up to the character limit).

`description` (*required=True*, *type=char*, *max_limit=1000*)

A succinct paragraph describing the group.

`documentation` (*required=True*, *type=char*, *max_limit=1000*)

If you would like formatted text to appear in the schema browser, please supply the name of the file containing html-formatted text (see *Formatting* for more info). Note, this is typically for 2-3 paragraphs of information. Detailed documentation should be written into a Document Central article. If you do not wish to supply documentation for a particular row, leave this entry blank.

`contact` (*required=True*, *type=char*, *max_limit=500*)

Format as: John Smith <john.smith@institute.org>

`date` (*required=True*, *type=char*, *max_limit=100*)

Group creation/update date as defined by the team e.g., 2012-04-23

`version` (*required=True*, *type=char*, *max_limit=100*)

Group version as defined by the team e.g., v1.8

<survey>_<datarelease>_table_meta.txt

This file describes the `tables` you would like to register, and will be used to populate the Schema Browser and be available for public SQL/ADQL querying, as well as discoverable through the Data Central TAP server.

Please provide a single .txt file with an entry per table, containing the following meta information:

name	description	documentation	group	filename	contact	date	version
ApMatchedCat	This table contains r-band aperture matched photometry and otherSource Extractor outputs for all GAMA DR2 objects.	unique_table_documentation_filename.txt	ApMatchedPhotom	ApMatchedCat.fits	name <email@institute.org>	2012-04-23	v02

Please name this file: <survey>_<datarelease>_table_meta.txt e.g., sami_dr2_table_meta.txt

`<survey>_<datarelease>_table_meta.txt`

This file should contain the following columns

`name` (*required=True*, *type=char*, *max_limit=100*)
Table name. Use only alphanumeric characters. This must be unique per data release.

`description` (*required=True*, *type=char*, *max_limit=1000*)
A succinct paragraph describing the group.

`documentation` (*required=True*, *type=char*, *max_limit=1000*)
If you would like formatted text to appear in the schema browser, please supply the name of the file containing html-formatted text (see *Formatting* for more info). Note, this is typically for 2-3 paragraphs of information. Detailed documentation should be written into a Document Central article. If you do not wish to supply documentation for a particular row, leave this entry blank.

`group_name` (*required=True*, *type=char*, *max_limit=100*)
The name of the group (must match a group name from the *<survey>_<datarelease>_group_meta.txt* file above)

`filename` (*required=True*, *type=char*, *max_limit=1000*)
The filename of the table you'll be providing

`contact` (*required=True*, *type=char*, *max_limit=500*)
Format as: John Smith <john.smith@institute.org>

`date` (*required=True*, *type=char*, *max_limit=100*)
Table creation/update date as defined by the team e.g., 2012-04-23

`version` (*required=True*, *type=char*, *max_limit=100*)
Table version as defined by the team e.g., v1.8

<survey>_<datarelease>_column_meta.txt

This file describes the `columns` you would like to register for each `table`, and will be used to populate the Schema Browser, SQL/ADQL query service, and the TAP server. Please provide the following a **single** pipe-delimited .txt file containing an entry (row) for each column:

name	table_name	description	ucd	unit	data_type
ALPHA_J2000	ApMatchedCat	RA (r band)	pos.eq.ra;em.opt.R	deg	double
CATAID	EnvironmentMeasures	Unique GAMA ID	meta.id	•	double

Please name this file: <survey_dr>_column_meta.txt e.g., sami_dr2_column_meta.txt

<survey>_<datarelease>_column_meta.txt

This file should contain the following columns

name (*required=True*, *type=char*, *max_limit=100*)

Column name. Use only alphanumeric characters.

> **Attention:** Column names must be SQL-queriable, use only characters, letters and underscores in your column names. Column names cannot start with numbers but can include numbers afterwards. Forbidden characters include: %^&({}+-/][‘”

description (*required=True*, *type=char*, *max_limit=1000*)

A succinct paragraph describing the table.

table_name (*required=True*, *type=char*, *max_limit=100*)

The name of the table (must match a table name from the *<survey>_<datarelease>_table_meta.txt* file above)

ucd (*required=True*, *type=char*, *max_limit=100*)

UCDs can be found here: http://cds.u-strasbg.fr/UCD/tree/js/ (more info: https://arxiv.org/pdf/1110.0525.pdf)

unit (*required=True*, *type=char*, *max_limit=100*)

Column unit

data_type (*required=True*, *type=char*, *max_limit=100*)

data type of the column. Add the full name of the data type such as integer instead of shorten form int.

2.1.4 Extra Requirements

Note: this section is optional, you do not have to provide an _coordinate_meta.txt file or _sql_meta.txt file if your data release does not lend itself to individual source identification.

You cannot ingest individual data products associated with a single astronomical object without completing this step.

By providing a metadata file pointing to the input catalogue of your data release, Data Central is able to populate the database with Astronomical Objects from your survey. These objects are then accessible in the name resolver, and cone search (as well as the image cutout overplotting functionality).

Source Catalogue Identification

To identify a table as a source catalogue (i.e. an input catalogue that has one row per source in your data release), please provide a metadata file that contains the name of a single table that contains the resolver info (source name, coordinates, format), as per:

table_name	source_name_col	long_col	lat_col	long_format	lat_format	frame	equinox
InputCatA	CATAID	RA_deg	Dec_deg	deg	deg	icrs	

Please name this file: <survey_dr>_coordinate_meta.txt e.g., sami_dr2_coordinate_meta.txt

> **Danger:** If your input table is > 2GB in size, please ensure the format is .csv (not fits).

Tip: It is advised to provide coordinates as RA, Dec (degrees, degrees). If your Long/Lat fields are not in an ICRS coordinate frame (degrees), Data Central will auto-generate these columns.

`<survey_dr>_coordinate_meta.txt`

This file should contain the following columns

`table_name` (*required=True*, *type=char*, *max_limit=100*)

The table name (not filename) to be used (must have an entry in the *<survey>_<datarelease>_table_meta.txt* file.)

`source_name_col` (*required=True*, *type=char*, *max_limit=100*)

The column name for source name (from the specified table)

`long_col` (*required=True*, *type=char*, *max_limit=100*)

The column name for longitude (from the specified table)

`lat_col` (*required=True*, *type=char*, *max_limit=100*)

The column name for latitude (from the specified table)

`long_format` (*required=True*, *type=char*, *max_limit=100*)

The longitude format. Depending on the formatting of your coordinate values (i.e., whether decimal/space delimited/colon delimited) and the value of long_format/lat_format (deg or h), coordinate data are interpreted as:

value	format	interpretation
10.2345	deg	Degrees
1 2 3	deg	Degrees, arcmin, arcsecond
1:2:30.40	deg	Sexagesimal degrees
1 2 0	hourangle	Sexagesimal hours

`lat_format` (*required=True*, *type=char*, *max_limit=100*)

The latitude format. Depending on the formatting of your coordinate values (i.e., whether decimal/space delimited/colon delimited) and the value of long_format/lat_format (deg or h), coordinate data are interpreted as:

value	format	interpretation
10.2345	deg	Degrees
1 2 3	deg	Degrees, arcmin, arcsecond
1:2:30.40	deg	Sexagesimal degrees
1 2 0	hourangle	Sexagesimal hours

`frame` (*required=True*, *type=char*, *max_limit=100*)

Coordinate frame. Accepted values are (fk5, fk4, icrs, galactic, supergalactic)

`equinox` (*required=True*, *type=char*, *max_limit=100*)

If appropriate (leave blank for icrs), the equinox of this frame. Accepted values are (j2000, j1950, b1950)

Data Central will auto-generate ICRS-frame RA(deg) Dec(deg) columns if that format has not been provided. Data Central is able to transform using the combinations of coordinate systems listed above. If you do not see the coordinate system your data are currently recorded in, it is advised to generate RA, Dec columns as ICRS for your catalogue to be included.

Danger: The values in `source_name_col` must be unique across all of the tables included in the source catalogue. If none of your existing tables meet this requirement, then you will need to generate a new table, which need only include source name, RA and Dec. You do not need to include this table in the column or table metadata files, but it would be preferred.

SOV SQL Functionality

If you wish for a your tables to be queried and rows displayed as part of the Single Object Viewer, please provide the following table:

table_name	sql
InputCatA	“SELECT * FROM gama_dr2.InputCatA WHERE CATAID = {objid}”

Please name this file: <survey_dr>_sql_meta.txt e.g., sami_dr2_sql_meta.txt

`<survey_dr>_sql_meta.txt`

This file should contain the following columns

`table_name` (*required=True*, *type=char*, *max_limit=100*)

The table name (not filename) to be used (must have an entry in the <survey_dr>_table_meta.txt file.)

`sql` (*required=True*, *type=char*, *max_limit=100*)

The SQL expression (following the DC syntax of survey_dr.table_name) to be run on SOV load. {objid} will be replaced by the AstroObject being requested.

Attention: Ensure your SQL runs before submitting this metadata file. e.g., check whether you need single quotes around the objid as per: ‘{objid}’

2.2 IFS

Ingesting IFS data products into Data Central allows for population of the Single Object Viewer, from which users are able to download and explore (file type dependent) those data products. The Schema Browser will also include an entry for each IFS data product type. The minimum requirements for ingesting IFS data products are described below.

Note: Please Note: this is a living document, additional requirements for various data product visualizations (2D maps etc) will be set, and updated, in the near future. Although Data Central will take every effort to minimize changes to the format of the meta-data required, in order to provide survey teams with interactive visualizations and deliver the best possible user experience, we may from time-to-time require changes to these requirements.

For ingested surveys, where possible, we will auto-generate any meta data required (from fits headers), but teams should be aware that this document will likely contain additional/updated requirements on a ~6 monthly timescale in order to be compatible with the latest Data Central release.

Important: Prerequisite: You’ll need to have provided an input catalogue of sources, as per the *Catalogues* documentation, following all sections up to and including *Source Catalogue Identification*.

Note: Remember that the documentation mentioned here is the static, paper-like documentation, the documentation on Documentent Central is entirely separate.

2.2.1 Data Model

Data Central's ingestion process will map your data onto the Data Central data model format. IFS data are organised hierarchically, as per:

```
Survey
└── DataRelease
    └── Schema:IFS
        └── Facility
            └── Data Product
```

To explore the data model further, visit the IFS section of the Schema Browser to explore the relationships between facilities and data products.

2.2.2 Directory Structure

To ingest IFS data products, you will provide two folders, one containing the data products themselves, and one containing the metadata. You'll also need to have provided an input catalogue of sources, as per the *Catalogues* documentation, following all sections up to and including *Source Catalogue Identification*.

Data

The ifs directory should contain the fits files themselves.

```
data
└── <survey>
    └── <datarelease>
        └── ifs
            └── <source_name>
                ├── product1.fits
                └── product2.gz
            └── <source_name>
                ├── product3.fits
                └── product4.fits
```

Attention: <survey> and <datarelease> should be replaced with the values you chose in *Getting Started*, e.g., sami and dr2

A good rule of thumb is to keep your files succinct and with as few extensions as possible (i.e, do not pack tens of extensions into your IFS cubes, separate into different files). This makes mapping the correct file (by data type) to a browser visualizer simpler. Currently we support 2D map visualizations in the browser, have a look at a SAMI DR2 object in the Single Object Viewer.

Metadata

The following file structure should be adopted. A top-level <survey> directory should contain a single directory per `<datarelease>`. Both directories should have metadata files described below which will populate the Schema Browser.

```
metadata
└── <survey>
    ├── <survey>_survey_meta.txt
    └── <datarelease>
        ├── <survey>_<data_release>_data_release_meta.txt
        └── ifs/
            ├── <survey>_<datarelease>_facility_meta.txt
            ├── <survey>_<datarelease>_group_meta.txt
            ├── <survey>_<datarelease>_product_ingestion.txt
            ├── <survey>_<datarelease>_product_meta.txt
            └── docs/
```

The metadata/catalogues/ directory will contain a minimum of 4 metadata files, plus a docs/ directory if you have supplied additional documentation for a particular facility/product.

2.2.3 Metadata Files

Attention: Metadata files are always pipe-delimited, and have the extension .txt

<survey>_<datarelease>_facility_meta.txt

This file describes the `facilities` you would like to register, and will be used to populate the Schema Browser and SOV. Provide the following a single pipe-delimited .txt file containing an entry (row) for each facility:

name	pretty_name	description | documentation
sami	SAMI	Brief SAMI instrument description | sami_instrument_doc.txt

Please name this file: <survey>_<datarelease>_facility_meta.txt e.g., sami_dr2_facility_meta.txt

`<survey>_<datarelease>_facility_meta.txt`
This file should contain the following columns

`name` (*required=True*, *type=char*, *max_limit=100*)

Facility name. Use only alphanumeric characters. This must be unique per data release.

`pretty_name` (*required=True*, *type=char*, *max_limit=100*)
A human-readable version of the facility name. This can contain any characters (up to the character limit).

`description` (*required=True*, *type=char*, *max_limit=1000*)
A succinct paragraph describing the facility.

`documentation` (*required=True*, *type=char*, *max_limit=1000*)
If you would like formatted text to appear in the schema browser, please supply the name of the file containing html-formatted text (see *Formatting* for more info). Note, this is typically for 2-3 paragraphs of information. Detailed documentation should be written into a Document Central article. If you do not wish to supply documentation for a particular row, leave this entry blank.

<survey>_<datarelease>_product_meta.txt

This file describes the types of `data products` you would like to register, and will be used to populate the Schema Browser and SOV. Provide the following a single pipe-delimited .txt file containing an entry (row) for each product:

facility_name	name	description	documentation	group_name	version	contact
sami	spectral_cube-blue	Fully reduced and flux calibrated SAMI cube (blue)	spec_cube_blue.html	cube	1.2	John Smith <john.smith@institute.org>
sami	sfr_default_1-comp	1-component star formation rate map from default cube	sfr.html	sfr	1.2	John Smith <john.smith@institute.org>

Please name this file: <survey>_<datarelease>_product_meta.txt e.g., sami_dr2_product_meta.txt

`<survey>_<datarelease>_product_meta.txt`

This file should contain the following columns

`name` (*required=True*, *type=char*, *max_limit=100*)

Data product name. Choose from the following:

name	pretty_name
cube_blue	Flux Cube: Blue
adaptive_blue	Adaptively-binned Flux Cube: Blue
annular_blue	Annularly-binned Flux Cube: Blue
sectors_blue	Sectors-binned Flux Cube: Blue
cube_red	Flux Cube: Red
adaptive_red	Adaptively-binned Flux Cube: Red
annular_red	Annularly-binned Flux Cube: Red
sectors_red	Sectors-binned Flux Cube: Red
spectrum_3-kpc_blue	3kpc Aperture Spectrum: Blue
spectrum_re_blue	Re Elliptical Aperture Spectrum: Blue
spectrum_1-4-arcsec_blue	1”.4 Aperture Spectrum: Blue
spectrum_2-arcsec_blue	2” Aperture Spectrum: Blue
spectrum_3-arcsec_blue	3” Aperture Spectrum: Blue
spectrum_4-arcsec_blue	4” Aperture Spectrum: Blue
spectrum_3-kpc_red	3kpc Aperture Spectrum: Red
spectrum_re_red	Re Elliptical Aperture Spectrum: Red
spectrum_1-4-arcsec_red	1”.4 Aperture Spectrum: Red
spectrum_2-arcsec_red	2” Aperture Spectrum: Red
spectrum_3-arcsec_red	3” Aperture Spectrum: Red
spectrum_4-arcsec_red	4” Aperture Spectrum: Red
stellar-velocity_default_two-moment	Stellar Velocity Map
stellar-velocity-dispersion_default_two-moment	Stellar Velocity Dispersion Map
stellar-velocity_adaptive_two-moment	Stellar Velocity Map from Adaptively-binned Cube
stellar-velocity-dispersion_adaptive_two-moment	Stellar Velocity Dispersion Map from Adaptively-binned Cube
stellar-velocity_annular_two-moment	Stellar Velocity Map from Annularly-binned Cube
stellar-velocity-dispersion_annular_two-moment	Stellar Velocity Dispersion Map from Annularly-binned Cube
stellar-velocity_sectors_two-moment	Stellar Velocity Map from Sectors-binned Cube
stellar-velocity-dispersion_sectors_two-moment	Stellar Velocity Dispersion Map from Sectors-binned Cube
gas-velocity_default_1-comp	1-component Ionised Gas Velocity Map
gas-vdisp_default_1-comp	1-component Ionised Gas Velocity Dispersion Map
OII3728_default_1-comp	1-component Line Emission Map: [OII] (3726Å+3729Å)

Table 1 – continued from previous page

Halpha_default_1-comp	1-component Line Emission Map: H
Hbeta_default_1-comp	1-component Line Emission Map: H
OIII5007_default_1-comp	1-component Line Emission Map: [OIII] (5007Å)
OI6300_default_1-comp	1-component Line Emission Map: [OI] (6300Å)
NII6583_default_1-comp	1-component Line Emission Map: [NII] (6583Å)
SII6716_default_1-comp	1-component Line Emission Map: [SII] (6716Å)
SII6731_default_1-comp	1-component Line Emission Map: [SII] (6731Å)
gas-velocity_default_recom-comp	Recommended-component Ionised Gas Velocity Map
gas-vdisp_default_recom-comp	Recommended-component Ionised Gas Velocity Dispersion Map
OII3728_default_recom-comp	Recommended-component Line Emission Map: [OII] (3726Å+3729Å)
Halpha_default_recom-comp	Recommended-component Line Emission Map: H
Hbeta_default_recom-comp	Recommended-component Line Emission Map: H
OIII5007_default_recom-comp	Recommended-component Line Emission Map: [OIII] (5007Å)
OI6300_default_recom-comp	Recommended-component Line Emission Map: [OI] (6300Å)
NII6583_default_recom-comp	Recommended-component Line Emission Map: [NII] (6583Å)
SII6716_default_recom-comp	Recommended-component Line Emission Map: [SII] (6716Å)
SII6731_default_recom-comp	Recommended-component Line Emission Map: [SII] (6731Å)
gas-velocity_adaptive_1-comp	1-component Ionised Gas Velocity Map from Adaptively-binned Cube
gas-vdisp_adaptive_1-comp	1-component Ionised Gas Velocity Dispersion Map from Adaptively-binned Cube
OII3728_adaptive_1-comp	1-component Line Emission Map from Adaptively-binned Cube: [OII] (3726Å+3
Halpha_adaptive_1-comp	1-component Line Emission Map from Adaptively-binned Cube: H
Hbeta_adaptive_1-comp	1-component Line Emission Map from Adaptively-binned Cube: H
OIII5007_adaptive_1-comp	1-component Line Emission Map from Adaptively-binned Cube: [OIII] (5007Å)
OI6300_adaptive_1-comp	1-component Line Emission Map from Adaptively-binned Cube: [OI] (6300Å)
NII6583_adaptive_1-comp	1-component Line Emission Map from Adaptively-binned Cube: [NII] (6583Å)
SII6716_adaptive_1-comp	1-component Line Emission Map from Adaptively-binned Cube: [SII] (6716Å)
SII6731_adaptive_1-comp	1-component Line Emission Map from Adaptively-binned Cube: [SII] (6731Å)
gas-velocity_annular_2-comp	2-component Ionised Gas Velocity Map from Annularly-binned Cube
gas-vdisp_annular_2-comp	2-component Ionised Gas Velocity Dispersion Map from Annularly-binned Cube
OII3728_annular_2-comp	2-component Line Emission Map from Annularly-binned Cube: [OII] (3726Å+37
Halpha_annular_2-comp	2-component Line Emission Map from Annularly-binned Cube: H
Hbeta_annular_2-comp	2-component Line Emission Map from Annularly-binned Cube: H
OIII5007_annular_2-comp	2-component Line Emission Map from Annularly-binned Cube: [OIII] (5007Å)
OI6300_annular_2-comp	2-component Line Emission Map from Annularly-binned Cube: [OI] (6300Å)
NII6583_annular_2-comp	2-component Line Emission Map from Annularly-binned Cube: [NII] (6583Å)
SII6716_annular_2-comp	2-component Line Emission Map from Annularly-binned Cube: [SII] (6716Å)
SII6731_annular_2-comp	2-component Line Emission Map from Annularly-binned Cube: [SII] (6731Å)
gas-velocity_sectors_1-comp	1-component Ionised Gas Velocity Map from Sectors-binned Cube
gas-vdisp_sectors_1-comp	1-component Ionised Gas Velocity Dispersion Map from Sectors-binned Cube
OII3728_sectors_1-comp	1-component Line Emission Map from Sectors-binned Cube: [OII] (3726Å+3729
Halpha_sectors_1-comp	1-component Line Emission Map from Sectors-binned Cube: H
Hbeta_sectors_1-comp	1-component Line Emission Map from Sectors-binned Cube: H
OIII5007_sectors_1-comp	1-component Line Emission Map from Sectors-binned Cube: [OIII] (5007Å)
OI6300_sectors_1-comp	1-component Line Emission Map from Sectors-binned Cube: [OI] (6300Å)
NII6583_sectors_1-comp	1-component Line Emission Map from Sectors-binned Cube: [NII] (6583Å)
SII6716_sectors_1-comp	1-component Line Emission Map from Sectors-binned Cube: [SII] (6716Å)
SII6731_sectors_1-comp	1-component Line Emission Map from Sectors-binned Cube: [SII] (6731Å)
gas-velocity_adaptive_recom-comp	Recommended-component Ionised Gas Velocity Map from Adaptively-binned Cu
gas-vdisp_adaptive_recom-comp	Recommended-component Ionised Gas Velocity Dispersion Map from Adaptively

Table 1 – continued from previous page

OII3728_adaptive_recom-comp	Recommended-component Line Emission Map from Adaptively-binned Cube: [O
Halpha_adaptive_recom-comp	Recommended-component Line Emission Map from Adaptively-binned Cube: H
Hbeta_adaptive_recom-comp	Recommended-component Line Emission Map from Adaptively-binned Cube: H
OIII5007_adaptive_recom-comp	Recommended-component Line Emission Map from Adaptively-binned Cube: [O
OI6300_adaptive_recom-comp	Recommended-component Line Emission Map from Adaptively-binned Cube: [O
NII6583_adaptive_recom-comp	Recommended-component Line Emission Map from Adaptively-binned Cube: [N
SII6716_adaptive_recom-comp	Recommended-component Line Emission Map from Adaptively-binned Cube: [S
SII6731_adaptive_recom-comp	Recommended-component Line Emission Map from Adaptively-binned Cube: [S
extinct-corr_default_1-comp	Extinction correction map from 1-component H/H flux ratio
sfr-mask_default_1-comp	Star formation mask map from 1-component emission line ratios
sfr_default_1-comp	Star formation rate map from 1-component H flux
sfr-dens_default_1-comp	Star formation rate surface density from 1-component H flux
extinct-corr_default_recom-comp	Extinction correction map from recommended-component H/H flux ratio
sfr-mask_default_recom-comp	Star formation mask map from recommended-component emission line ratios
sfr_default_recom-comp	Star formation rate map from recommended-component H flux
sfr-dens_default_recom-comp	Star formation rate surface density from recommended-component H flux
gas-velocity_sectors_recom-comp	Recommended-component Ionised Gas Velocity Map from Sectors-binned Cube
gas-vdisp_sectors_recom-comp	Recommended-component Ionised Gas Velocity Dispersion Map from Sectors-bi
OII3728_sectors_recom-comp	Recommended-component Line Emission Map from Sectors-binned Cube: [OII]
Halpha_sectors_recom-comp	Recommended-component Line Emission Map from Sectors-binned Cube: H
Hbeta_sectors_recom-comp	Recommended-component Line Emission Map from Sectors-binned Cube: H
OIII5007_sectors_recom-comp	Recommended-component Line Emission Map from Sectors-binned Cube: [OIII
OI6300_sectors_recom-comp	Recommended-component Line Emission Map from Sectors-binned Cube: [OI] (
NII6583_sectors_recom-comp	Recommended-component Line Emission Map from Sectors-binned Cube: [NII]
SII6716_sectors_recom-comp	Recommended-component Line Emission Map from Sectors-binned Cube: [SII]
SII6731_sectors_recom-comp	Recommended-component Line Emission Map from Sectors-binned Cube: [SII]
extinct-corr_adaptive_1-comp	Extinction correction map from 1-component adaptive-binned H/H flux ratio
sfr-mask_adaptive_1-comp	Star formation mask map from 1-component adaptive-binned emission line ratios
sfr_adaptive_1-comp	Star formation rate map from 1-component adaptive-binned H flux
sfr-dens_adaptive_1-comp	Star formation rate surface density from 1-component adaptive-binned H flux
extinct-corr_adaptive_recom-comp	Extinction correction map from recommended-component adaptive-binned H/H f
sfr-mask_adaptive_recom-comp	Star formation mask map from recommended-component adaptive-binned emissi
sfr_adaptive_recom-comp	Star formation rate map from recommended-component adaptive-binned H flux
sfr-dens_adaptive_recom-comp	Star formation rate surface density from recommended-component adaptive-binn
extinct-corr_sectors_1-comp	Extinction correction map from 1-component sectors-binned H/H flux ratio
sfr-mask_sectors_1-comp	Star formation mask map from 1-component sectors-binned emission line ratios
sfr_sectors_1-comp	Star formation rate map from 1-component sectors-binned H flux
sfr-dens_sectors_1-comp	Star formation rate surface density from 1-component sectors-binned H flux
extinct-corr_sectors_recom-comp	Extinction correction map from recommended-component sectors-binned H/H flu
sfr-mask_sectors_recom-comp	Star formation mask map from recommended-component sectors-binned emissio
sfr_sectors_recom-comp	Star formation rate map from recommended-component sectors-binned H flux
sfr-dens_sectors_recom-comp	Star formation rate surface density from recommended-component sectors-binned
extinct-corr_annular_2-comp	Extinction correction map from 2-component annular-binned H/H flux ratio
sfr-mask_annular_2-comp	Star formation mask map from 2-component annular-binned emission line ratios
sfr_annular_2-comp	Star formation rate map from 2-component annular-binned H flux
sfr-dens_annular_2-comp	Star formation rate surface density from 2-component annular-binned H flux

If you have additional product names that aren't covered by this list, please let us know as soon as possible.

facility_name (*required=True*, *type=char*, *max_limit=100*)

The name of the facility (must match a facility name from the *<survey>_<datarelease>_facility_meta.txt* file)

description (*required=True*, *type=char*, *max_limit=1000*)
A succinct paragraph describing the product.

documentation (*required=True*, *type=char*, *max_limit=1000*)
If you would like formatted text to appear in the schema browser, please supply the name of the file containing html-formatted text (see *Formatting* for more info). Note, this is typically for 2-3 paragraphs of information. Detailed documentation should be written into a Document Central article. If you do not wish to supply documentation for a particular row, leave this entry blank.

group_name (*required=True*, *type=char*, *max_limit=100*)
The name of the group (must match a group name from the *<survey>_<datarelease>_group_meta.txt* file)

version (*required=True*, *type=char*, *max_limit=100*)
Product version as defined by the team e.g., v1.8

contact (*required=True*, *type=char*, *max_limit=500*)
Format as: John Smith <john.smith@institute.org>

<survey>_<datarelease>_group_meta.txt

To aid the user in finding data products, a grouping is applied in the Schema Browser and the SOV (i.e., all IFS-derived spectra can be grouped under 'Spectra'). This file describes the `groups` you would like to register, and will be used to populate the Schema Browser and categorize data products in the SOV. Provide the following a single pipe-delimited .txt file containing an entry (row) for each group:

name	pretty_name
cube	Cubes
stelkin	Stellar Kinematics

Please name this file: <survey>_<datarelease>_group_meta.txt e.g., sami_dr2_group_meta.txt

<survey>_<datarelease>_group_meta.txt
This file should contain the following columns

name (*required=True*, *type=char*, *max_limit=100*)
Group name. Use only alphanumeric characters. This must be unique per data release.

pretty_name (*required=True*, *type=char*, *max_limit=100*)
A human-readable version of the group name. This can contain any characters (up to the character limit).

<survey>_<datarelease>_product_ingestion.txt

The final file you will need to provide links each data product with the relevant source from your input catalogue (as mentioned above, you'll need to have provided an input catalogue of sources, as per the *Catalogues* documentation, following all sections up to and including *Source Catalogue Identification*). This file should include an entry (row) for every file you wish to ingest into Data Central.

facility_name	data_product_name	file_name	rel_file_path	source_name
sami	OIII5007_annular_2-comp	99428_OIII5007_annular_2-comp.fits	99428/99428_OIII5007_annular_2-comp.fits	99428

Please name this file: <survey_dr>_product_ingestion.txt e.g., sami_dr2_product_ingestion.txt

`<survey>_<datarelease>_product_ingestion.txt`

This file should contain the following columns

`facility_name` (*required=True*, *type=char*, *max_limit=100*)

The name of the facility (must match a facility name from the *<survey>_<datarelease>_facility_meta.txt* file)

`data_product_name` (*required=True*, *type=char*, *max_limit=100*)

The name of the data product (must match a facility name from the *<survey>_<datarelease>_product_meta.txt* file)

`file_name` (*required=True*, *type=char*, *max_limit=100*)

The filename of the data product you'll be providing

`rel_file_path` (*required=True*, *type=char*, *max_limit=100*)

The relative path of the file. e.g., <source_name>/product1.fits. See *Directory structure*.

`source_name` (*required=True*, *type=char*, *max_limit=100*)

The source name as provided in your source catalogue (see *Source Catalogue Identification*).

`is_best` (*required=False*, *type=bool*, *default=True*)

You may associate multiple IFS with a single product, and define ONE of those products as the *best* (add a description of this in your *<survey>_<datarelease>_product_meta.txt* file or data release information on Document Central. The *best* product for a source will be highlighted in the SOV as such.

2.3 Images

Ingesting fits images allows for their retrieval via the a multi-band image cutout service. This service allows astronomers to easily and intuitively access imaging data from some of the world's leading facilities alongside datasets of national significance, to quickly explore sources of interest and produce publication quality images.

The image cutout service allows for comparison of the same source at different wavelengths side by side. Users can input a single source/position or a list (automatically transforming between the submitted coordinate frame and the native coordinate frame of the image) to generate single bands and/or three-colour images.

Five image scaling options are supported, along with custom plot options (axes style, colour map, coordinate grid), over-plotting source positions from any of the DC-hosted input catalogues as well as remote VO datasets, and over-plotting contours for a particular band at user-specified levels. Users are able to download results as .png or .fits files, on a per-band/per-source basis, or download all the files from one request.

The cutout can host mosaiced image files in a range of projections and pixel scales, automatically reprojecting data on the fly.

Note: Remember that the documentation mentioned here is the static, paper-like documentation, the documentation on Documentent Central is entirely separate.

2.3.1 Data Model

Data Central's ingestion process will map your data onto the Data Central data model format. Within Data Central, imaging files are organised hierarchically, as per:

```
Survey
└── DataRelease
    └── Schema:Imaging
```

(continues on next page)

(continued from previous page)

```
        └── Facility
            └── Band
                └── Region
```

Currently, the cutout service is not able to stitch together imaging on the fly. You must provide FITS images that have been resampled and co-added (see for example, SWarp.) Each file is saved as a `Region` on the sky, with the fits header stored in the database to provide the coordinate lookup. Note, this means individual region files must uniquely cover one area of sky. Data central can not currently combine or display multiple images in the same band covering the sam sky region. To explore the data model further, visit the imaging section of the Schema Browser to explore the relationships between facilities and bands.

2.3.2 Directory Structure

To ingest data into Data Central, you will provide two folders, one containing the imaging files themselves, and one containing the metadata.

Data

The imaging directory should contain the fits files themselves.

```
data
└── <survey>
    └── <datarelease>
        └── imaging
            ├── swpim_galex_FUV_d10.fits
            └── swpim_galex_NUV_g09.fits
```

Attention: `<survey>` and `<datarelease>` should be replaced with the values you chose in *Getting Started*, e.g., gama and dr2

Data Central supports images in .fits formats. Note, these data files can technically have any file name, as long as they are correctly matching in your metadata files (see below). However, we strongly suggest using sensible file names that describe the facility, band, survey and region.

Metadata

The following file structure should be adopted. A top-level `<survey>` directory should contain a single directory per `<datarelease>`. Both directories should have metadata files described below which will populate the Schema Browser.

```
metadata
└── <survey>
    ├── <survey>_survey_meta.txt
    └── <datarelease>
        ├── <survey>_<data_release>_data_release_meta.txt
        └── imaging/
            ├── <survey>_<datarelease>_facility_meta.txt
            ├── <survey>_<datarelease>_band_meta.txt
            └── docs/
```

The metadata/imaging/ directory will contain 2 metadata files, plus a docs/ directory if you have supplied additional documentation for a particular facility/band.

2.3.3 Metadata Files

Attention: Metadata files are always pipe-delimited, and have the extension .txt

<survey>_<datarelease>_facility_meta.txt

This file describes the `facilities` you would like to register, and will be used to populate the Schema Browser. Provide the following a single pipe-delimited .txt file containing an entry (row) for each facility:

name	pretty_name	description	documentation
CFHTCFHTCOSMOS	CFHT	Images from the CFHT COSMOS survey.	unique_cfht_documentation_filename.txt

Please name this file: <survey>_<datarelease>_facility_meta.txt e.g., devils_dr1_facility_meta.txt

`<survey>_<datarelease>_facility_meta.txt`
This file should contain the following columns

`name` (*required=True*, *type=char*, *max_limit=100*)
Facility name. Use only alphanumeric characters. This must be unique per data release.

`pretty_name` (*required=True*, *type=char*, *max_limit=100*)
A human-readable version of the facility name. This can contain any characters (up to the character limit).

`description` (*required=True*, *type=char*, *max_limit=1000*)
A succinct paragraph describing the facility.

`documentation` (*required=True*, *type=char*, *max_limit=1000*)
If you would like formatted text to appear in the schema browser, please supply the name of the file containing html-formatted text (see *Formatting* for more info). Note, this is typically for 2-3 paragraphs of information. Detailed documentation should be written into a Document Central article. If you do not wish to supply documentation for a particular row, leave this entry blank.

<survey>_<datarelease>_band_meta.txt

This file describes the `bands` you would like to register, and will be used to populate the Schema Browser and be available to select in the cutout service.

Please provide a single .txt file with an entry per file (each file should have a corresponding unique 'region'), containing the following meta information:

name	pretty_name	start_wave_ang	central_wave_ang	end_wave_ang	spatial_res	start_time	end_time	pol_state	facility	description	documentation	contact	version	file_name	region	is_displayed
H	H	14629	16499.1	18085	0.5				CFHTCOSMOS	CFHTCOSMOS H-Band	unique_band_doc.txt	name <email@institute.org>	v02	swpim_cfht_H_d10.fits	d10	Y

Please name this file: <survey>_<datarelease>_band_meta.txt e.g., devils_dr1_band_meta.txt

`<survey>_<datarelease>_band_meta.txt`

This file should contain the following columns

`name` (*required=True*, *type=char*, *max_limit=100*)
Band name. This must be unique per data release.

`pretty_name` (*required=True*, *type=char*, *max_limit=100*)
A human-readable version of the band name. This can contain any characters (up to the character limit).

`start_wave_ang` (*required=True*, *type=char*, *max_limit=100*)
A wavelength for the start of the filter bandpass of the image - in Angstrom. This is required by the Simple Image Access service.

`central_wave_ang` (*required=True*, *type=char*, *max_limit=100*)
A rough central wavelength of the image - in Angstrom. This is used to order the bands in the cutout service options.

`end_wave_ang` (*required=True*, *type=char*, *max_limit=100*)
A wavelength for the end of the filter bandpass of the image - in Angstrom. This is required by the Simple Image Access service.

`spatial_res` (*required=True*, *type=char*, *max_limit=100*)
The spatial resolution (FWHM) of the image - in arcsec. This is required by the Simple Image Access service.

`start_time` (*required=True*, *type=char*, *max_limit=100*)
The start time of the image observation in UTC isoformat: yyyy-mm-ddThh:mm:ss.ss. In case of mosaics made of multiple images, this may either be left blank or preferably correspond to the start time of the earliest image in the mosaic. This is required by the Simple Image Access service.

`end_time` (*required=True*, *type=char*, *max_limit=100*)
The end time of the image observation in UTC isoformat: yyyy:mm:ddThh:mm:ss.ss. In case of mosaics made of multiple images, this may either be left blank or preferably correspond to the end time of the latest image in the mosaic. This is required by the Simple Image Access service.

`pol_state` (*required=True*, *type=char*, *max_limit=100*)
The polarization state(s) of the image. An empty string if irrelevant (e.g. optical data), otherwise a space-separated string consisting of one or more of the following: I Q U V RR LL RL LR XX YY XY YX POLI POLA. This is required by the Simple Image Access service.

`facility` (*required=True*, *type=char*, *max_limit=100*)
The name of the facility (must match a facility name from the *<survey>_<datarelease>_facility_meta.txt* file above)

`description` (*required=True*, *type=char*, *max_limit=1000*)
A succinct paragraph describing the facility.

`documentation` (*required=True*, *type=char*, *max_limit=1000*)
If you would like formatted text to appear in the schema browser, please supply the name of the file containing html-formatted text (see *Formatting* for more info). Note, this is typically for 2-3 paragraphs of information. Detailed documentation should be written into a Document Central article. If you do not wish to supply documentation for a particular row, leave this entry blank.

`contact` (*required=True*, *type=char*, *max_limit=500*)
Format as: John Smith <john.smith@institute.org>

`version` (*required=True*, *type=char*, *max_limit=100*)
Band version as defined by the team e.g., v1.8

`file_name` (*required=True*, *type=char*, *max_limit=1000*)
The filename of the fits file you'll be providing. Note, you can provide multiple files at the same wavelength by providing an entry per file.

`region` (*required=True*, *type=char*, *max_limit=1000*)
The name of the region for this file (e.g., g09, g12). You may choose to define this from the center RA of the image.

`is_displayed` (*required=True*, *type=bool*)
Y or N. If N, this row will be skipped on ingestion (you may wish to give us the meta data now and the file later).

Attention: The fits header must contain the following keywords:

Keyword	Description
GAIN	Maximum equivalent gain (e-/ADU)
CD1_1	Linear projection matrix
CD1_2	Linear projection matrix
CD2_1	Linear projection matrix
CD2_2	Linear projection matrix
NAXIS	Number of axes
CRPIX1	Reference pixel on this axis
CRPIX2	Reference pixel on this axis
CRVAL1	World coordinate on this axis
CRVAL2	World coordinate on this axis
CTYPE1	WCS projection type for this axis
CTYPE2	WCS projection type for this axis
CUNIT1	Axis unit
CUNIT2	Axis unit
NAXIS1	Number of elements along this axis
NAXIS2	Number of elements along this axis
EQUINOX	Mean equinox
RADECSYS	Astrometric system

2.4 Spectra

Ingesting spectral data products into Data Central allows for population of the Single Object Viewer, and enables download of those data product files. The Schema Browser will also include an entry for each spectral data product type. The minimum requirements for ingesting spectral data products are described below.

Note: Please Note: this is a living document, additional requirements for various data product visualizations (1D spectra etc) will be set, and updated, in the near future. Although Data Central will take every effort to minimize changes to the format of the meta-data required, in order to provide survey teams with interactive visualizations and deliver the best possible user experience, we may from time-to-time require changes to these requirements.

For ingested surveys, where possible, we will auto-generate any meta data required (from fits headers), but teams should be aware that this document will likely contain additional/updated requirements on a ~6 monthly timescale in order to be compatible with the latest Data Central release.

Important: Prerequisite: You'll need to have provided an input catalogue of sources, as per the *Catalogues* documentation, following all sections up to and including *Source Catalogue Identification*.

Note: Remember that the documentation mentioned here is the static, paper-like documentation, the documentation on Documentent Central is entirely separate.

2.4.1 Data Model

Data Central's ingestion process will map your data onto the Data Central data model format. Spectra are organised hierarchically, as per:

```
Survey
└── DataRelease
    └── Schema:Spectra
        └── Facility
            └── Data Product
```

To explore the data model further, visit a survey with spectra (e.g., GAMA DR2) in the Schema Browser to explore the relationships between facilities and data products.

2.4.2 Directory Structure

To ingest spectra, you will provide two folders, one containing the data products themselves, and one containing the metadata. You'll also need to have provided an input catalogue of sources, as per the *Catalogues* documentation, following all sections up to and including *Source Catalogue Identification*.

Data

The spectra directory should contain the fits files themselves.

```
data
└── <survey>
    └── <datarelease>
        └── spectra
            └── <facility_name>
                ├── product1.fits
                └── product2.gz
            └── <facility_name>
                ├── product3.fits
                └── product4.fits
```

Attention: <survey> and <datarelease> should be replaced with the values you chose in *Getting Started*, e.g., gama and dr2

Data Central supports spectra .fits format only (single file per spectrum).

A good rule of thumb is to keep your files succinct and with as few extensions as possible (i.e,, do not pack tens of extensions into your fits files). This makes mapping the correct file (by data type) to a browser visualizer simpler.

Metadata

The following file structure should be adopted. A top-level <survey> directory should contain a single directory per <datarelease>. Both directories should have metadata files described below which will populate the Schema Browser.

```
metadata
└── <survey>
    ├── <survey>_survey_meta.txt
    └── <datarelease>
        ├── <survey>_<data_release>_data_release_meta.txt
        └── spectra/
            ├── <survey>_<datarelease>_facility_meta.txt
            ├── <survey>_<datarelease>_product_ingestion.txt
            ├── <survey>_<datarelease>_product_meta.txt
            └── docs/
```

The metadata/catalogues/ directory will contain a minimum of 3 metadata files, plus a docs/ directory if you have supplied additional documentation for a particular facility/product.

2.4.3 Metadata Files

Attention: Metadata files are always pipe-delimited, and have the extension .txt

<survey>_<datarelease>_facility_meta.txt

Provide the following a single pipe-delimited txt file containing an entry (row) for each facility:

name	pretty_name	description | documentation
2dfgrs	2dFGRS	All 2dFGRS spectra of GAMA DR2 objects obtained through the 2dFGRS query page. | 2dFGRS_survey.html

Please name this file: <survey>_<datarelease>_facility_meta.txt e.g., gama_dr2_facility_meta.txt

`<survey>_<datarelease>_facility_meta.txt`

This file should contain the following columns

`name` (*required=True*, *type=char*, *max_limit=100*)

Facility name. Use only alphanumeric characters. This must be unique per data release.

`pretty_name` (*required=True*, *type=char*, *max_limit=100*)

A human-readable version of the facility name. This can contain any characters (up to the character limit).

`description` (*required=True*, *type=char*, *max_limit=1000*)

A succinct paragraph describing the facility.

`documentation` (*required=True*, *type=char*, *max_limit=1000*)

If you would like formatted text to appear in the schema browser, please supply the name of the file containing html-formatted text (see *Formatting* for more info). Note, this is typically for 2-3 paragraphs of information. Detailed documentation should be written into a Document Central article. If you do not wish to supply documentation for a particular row, leave this entry blank.

<survey>_<datarelease>_product_meta.txt

All types of data products (regardless of whether a particular astro object has the data product) will appear in the Schema Browser. Provide the following a single pipe-delimited txt file containing an entry (row) for each product:

facility_name	name	description	documentation	version	contact
2dfgrs	spectrum_1d	Reduced 1D spectrum	2dfgrs_spec.html	1.2	John Smith <john.smith@institute.org>
2qz	spectrum_1d	Reduced 1D spectrum		1.2	John Smith <john.smith@institute.org>

Please name this file: <survey_dr>_product_meta.txt e.g., gama_dr2_product_meta.txt

> **Attention:** Depending on the value for the **name** column, you will need to add additional columns to this table. See File Formats for more information.

<survey>_<datarelease>_product_meta.txt
This file should contain the following columns

name (*required=True*, *type=char*, *max_limit=100*)
Data product name. Choose from the following:

name	value	vis_type
spectrum_1d	1D Spectrum	1d_spectrum
spectrum_2d	2D Spectrum	2d_spectrum

Now check the File Formats section for any additional columns you may need to add to this file. If you have additional product names that aren't covered by this list, please let us know as soon as possible.

facility (*required=True*, *type=char*, *max_limit=100*)
The name of the facility (must match a facility name from the *<survey>_<datarelease>_facility_meta.txt* file)

description (*required=True*, *type=char*, *max_limit=1000*)
A succinct paragraph describing the product.

documentation (*required=True*, *type=char*, *max_limit=1000*)
If you would like formatted text to appear in the schema browser, please supply the name of the file containing html-formatted text (see *Formatting* for more info). Note, this is typically for 2-3 paragraphs of information. Detailed documentation should be written into a Document Central article. If you do not wish to supply documentation for a particular row, leave this entry blank.

version (*required=True*, *type=char*, *max_limit=100*)
Product version as defined by the team e.g., v1.8

contact (*required=True*, *type=char*, *max_limit=500*)
Format as: John Smith <john.smith@institute.org>

data_format (*required=True*, *type=char*, *max_limit=100*)
This describes the format of the data, and tells Data Central how to read the files from the facility. Note that all the files from a facility must be in the same format (you can create multiple facilities if needed).

This can be one of three options:

- *Single-Split*: the data is stored in multiple HDUs, with each HDU composed of a single 1D array.

- *Multiline-Single*: the data is stored in a single HDU, across multiple lines (i.e. a 2D array).
- *specutils*: the data is in neither of the above formats, and should be loaded via a specific **specutils_** loader. If you think this might be the case, please contact us as soon as possible, as this option is somewhat more complex than the first two.

If *specutils is selected*, then add additional column *specutils_format*, which is the name of the **specutils_** loader to use to read the file.

If *Multiline-Single* or *Single-Split* are selected, then add the additional columns *all_keywords*, *all_standard_units*, *valid_wcs* and (if needed, see below) *fixup_file*.

specutils_format (*required=True*, *type=char*, *max_limit=100*)
This is the name of the specutils loader to use to read the fits file. A list of possible loaders can be found here, and instructions for creating new loaders can be found here.

valid_wcs (*required=True*, *type=bool*)
Is the WCS valid (e.g. do the axes match what is expected, do they meet the WCS standard)? Either true or false. The best strategy to check this is to use astropy to try to read the WCS from the FITS header.

If false, *fixup_file* must be set.

all_standard_units (*required=True*, *type=bool*)
Are all the required units in the header, with the right keyword? Either true or false. In general, this means that both *CUNIT1* (for the spectral unit, e.g. wavelength or frequency) and *BUNIT* (for the flux unit) are set. Try reading these header keywords with astropy if you are unsure.

If false, *fixup_file* must be set.

all_keywords (*required=True*, *type=bool*)
Do the HDU(s) include the required keywords to determine their purpose? Either true or false.

We look at different keywords depending on *Single-Split* or *Multiline-Single*.

For *Multiline-Single*, either the *EXTNAME* or *HDUNAME* keyword is used, whereas for *Single-Split*, either the *ROW* or *ARRAY* keywords are used.

Using the keyword, the value of the keyword is matched to those in the table below (in a case-insensitive manner).

Value	Purpose
badpix	Skip this row/HDU, do not treat as a spectra.
blank value	Skip this row/HDU, do not treat as a spectra.
sky	Treat as sky spectra, matching to last science spectra read.
stdev	Treat as standard deviation on the last science spectra read.
sigma	Treat as standard deviation on the last science spectra read.
variance	Treat as variance on the last science spectra read.
spectrum	Treat as a science spectra.

These keywords are then mapped to one of the purposes below (the following can also be used as the value in the FITS file).

Name	Purpose
skip	Skip this HDU/row, don't try to interprate it as a spectrum (e.g. is an image or table).
science	The actual science spectrum.
error_*	The error on a science spectrum. It is assumed that this applies to the previous science spectrum. If a different scheme is needed, a specutils loader should be used. Currently the possible values are error-stdev, error-variance and error-inversevariance, which come from what error handling is supported by astropy. Other error schemes should add support to astropy, or mark the row/HDU as skip.
combined_*	A science or error spectrum combined from other spectra within the file. This will be used preferentially over other spectra found, so should only appear once.
sky	Spectra of the sky (presumably used for reduction purposes).
unreduced_*	Unreduced science or error, will preferentially not be used over other spectra, unless all spectra are unreduced (unlikely).

If false (or you would like to specify more detail about the purpose of each HDU or row), *fixup_file* must be set.

fixup_file (*required=True*, *type=char*, *max_limit=100*)
The fixup file is a YAML file which includes additional metadata about how to read and interpret your spectra, and the format is documented below. *fixup_file* is the relative path to this file. You can have one fixup file per facility.

The fixup file

The fixup file is a YAML file with a specific set of information which tells Data Central how to understand your spectra. YAML is common human readable and writeable configuration format (vs. JSON or similar which while readable are quite hard to write).

If you have not seen or written YAML before, https://camel.readthedocs.io/en/latest/yamlref.html is a good reference—we only use a small subset of the features of YAML, so don't feel you need to understand everything on that page.

There are three sections in the fixup file, *wcs*, *units* and *hdus*/*hdu*.

wcs sets which FITS header keywords or the values to use for WCS. The following keys are used within the fixup file (Note that either the value or keyword should be specified. Otherwise all keys are required.):

- *pixel_reference_point_keyword* or *pixel_reference_point*: The reference pixel to use, this is usually set by the FITS keyword *CRPIX1*.
- *pixel_reference_point_value_keyword* or *pixel_reference_point_value*: The value of the reference pixel, this is usually set by the FITS keyword *CRVAL1*.
- *pixel_width_keyword* or *pixel_width*: The width of each pixel in the WCS unit, this is usually set by the FITS keyword *CDELT1*.
- *wavelength_unit_keyword* or *wavelength_unit*: The units that the WCS values are in, this is usually set by the FITS keyword *CUNIT1*.

units sets which FITS header keywords or the values to use for the flux. The following keys are used within the fixup file (Note that either the value or keyword should be specified.):

- *flux_unit_keyword* or *flux_unit*: The units the spectral flux is in. This is usually set by the *BUNIT* FITS keyword. One of these keywords is required.

- *flux_scale_keyword* or *flux_scale*: How much to scale the flux by. This is usually set by the *BSCALE* FITS keyword, and if not specified is treated as 1.

The third and most complex section is the *hdus/hdu* section. *hdus* is used by the *Single-Split* format, whereas *hdu* is used by the *Multiline-Single* format. Both systems allow for quite precise specification of how the data should be treated. Ideally, this section isn't needed, and your FITS headers contain sufficient metadata that everything can be automatically detected. Also, this section can only do so much—if there needs to be different metadata specified for each individual FITS file, then this section is insufficient. In that case, creating a specutils loader might be the best way forward.

For the *hdu* section, you can specify some metadata for the whole HDU, and some additional data on a per-row basis. For the whole hdu, the following options are available:

- *require_transpose*: Defaults to False, this is if the data is stored in columns rather than rows.
- *purpose_prefix*: Which keywords hold the purpose of each row. Should be of the form <PREFIX>n e.g. ROW0 ROW1 etc. Must cover all rows. The valid values can be found under the *all_keywords* column.

For each row, in addition to the keywords under the *wcs* and *units* sections, the *purpose* key can be set, and follows the same rules as *all_keywords*.

The *hdus* sections is very similar to the *hdu* section, though rather than dealing with rows, it deals with HDUs. For the whole file, a *purpose_prefix* can be set (following the same rules given under the *hdu* section), and for each HDU a *purpose* key can be set (again using the same rules as the *hdu* section).

Finally, under the *hdu* or *hdus* section, a single *cycle* key can be set. This follows the same rules as a normal *hdu* or *hdus* section (other than it cannot contain another cycle), and allows specifying metadata when there may be a variable number of rows/HDUs with a single collective purpose. An example fixup files (from the OzDES survey) is below:

```yaml
hdus:
  "0":
    purpose: "combined_science"
  "1":
    purpose: "combined_error_variance"
  "2":
    purpose: "skip"
  "cycle":
    "0":
      purpose: "science"
    "1":
      purpose: "error_variance"
    "2":
      purpose: "skip"
```

<survey>_<datarelease>_product_ingestion.txt

The final file you will need to provide links each data product with the relevant source from your input catalogue (as mentioned above, you'll need to have provided an input catalogue of sources, as per the *Catalogues* documentation, following all sections up to and including *Source Catalogue Identification*). This file should include an entry (row) for every file you wish to ingest into Data Central.

facility_name	data_product	file_name	rel_file_path	data_format	group_name	is_best	hdu	purpose	ra	dec	wmin	wmax	start_time	end_time	z	hrv	target_name	snr	resolving_power	spatial_res
gama	spectrum_1d	G12_Y3_017_187.fit	[illegible]	[illegible]	[illegible]	TRUE	[illegible]	science	174.006720	[illegible]	[illegible]	[illegible]	2009-01-01T00:00:00Z	2009-01-01T00:00:00.00Z	0.05		[illegible]	[illegible]	[illegible]	[illegible]

Please name this file: <survey_dr>_product_ingestion.txt e.g., gama_dr2_product_ingestion.txt

<survey>_<datarelease>_product_ingestion.txt

This file should contain the following columns

facility_name (*required=True*, *type=char*, *max_limit=100*)
The name of the facility (must match a facility name from the *<survey>_<datarelease>_facility_meta.txt* file)

data_product_name (*required=True*, *type=char*, *max_limit=100*)
The name of the data product (must match a facility name from the *<survey>_<datarelease>_product_meta.txt* file)

file_name (*required=True*, *type=char*, *max_limit=100*)
The filename of the data product you'll be providing

rel_file_path (*required=True*, *type=char*, *max_limit=100*)
The relative path of the file. e.g., <facility_name>/product1.fits. See *Directory structure*.

source_name (*required=True*, *type=char*, *max_limit=100*)
The source name as provided in your source catalogue (see *Source Catalogue Identification*).

specid (*required=False*, *type=char*, *max_limit=60*)
A name/identifier used to distinguish between different spectra with the same `source_name`. Can include the source name, but that is not required.

is_best (*required=False*, *type=bool*, *default=True*)
You may associate multiple spectra with a single source, and define ONE of those spectra as the *best* (add a description of this in your *<survey>_<datarelease>_product_meta.txt* file or data release information on Document Central. The *best* spectra for a source will be highlighted in the SOV as such.

ra (*required=True*, *type=char*, *max_limit=30*)
The RA of the spectrum (icrs) in decimal degrees. This is required by the Simple Spectrum Access service.

dec (*required=True*, *type=char*, *max_limit=30*)
The Declination of the spectrum (icrs) in decimal degrees. This is required by the Simple Spectrum Access service.

Simple Spectral Access Metadata

In addition to the other metadata needed to load and read the spectra files, surveys need to provide metadata such that the spectra files can be served over the Simple Spectral Access Protocol (SSA). This metadata can either be in the individual FITS files (preferred), or provided by additional files as part of the metadata preparation process.

<survey>_<datarelease>_product_ingestion.txt

start_time (*required=True*, *type=char*, *max_limit=50*)
The start time of the observation in UTC isoformat: yyyy-mm-ddThh:mm:ss.ss. If the spectrum is the combination of multiple spectra, use the start time of the earliest spectrum. By default, the FITS keyword *DATE-OBS* will be looked at.

end_time (*required=True*, *type=char*, *max_limit=50*)
The end time of the observation in UTC isoformat: yyyy-mm-ddThh:mm:ss.ss. If the spectrum is the combination of multiple spectra, use the end time of the latest spectrum. By default, the FITS keyword *DATE-END* will be looked at.

wmin (*required=True*, *type=float*)
The start wavelength of the spectrum - in Angstrom. This should be computable via the WCS information within the file.

wmax (*required=True*, *type=float*)
The end wavelength of the spectrum - in Angstrom. This should be computable via the WCS information within the file.

z (*required=False*, *type=float*)
The determined redshift of the spectrum. By default, the FITS keyword *Z* will be looked at.

hrv (*required=False*, *type=float*)
The determined heliocentric radial velocity of the spectrum - in km/s. By default, the FITS keyword *TODO* will be looked at. TODO

target_name (*required=False*, *type=char*, *max_limit=150*)
The preferred target name of the spectrum as a string. By default, the FITS keyword *OBJECT* will be looked at.

snr (*required=False*, *type=float*)
A representative signal-to-noise ratio of the spectrum (e.g. median or at the central wavelength). By default, the FITS keyword *TODO* will be looked at. TODO

resolving_power (*required=False*, *type=float*)
The resolving power (lambda/delta lambda) of the spectrum at its central wavelength. By default, the FITS keyword *TODO* will be looked at. TODO

spatial_res (*required=False*, *type=float*)
The spatial resolution corresponding to the PSF of the observed spectrum - in arcsec. By default, the FITS keyword *TODO* will be looked at. TODO

If a different FITS keyword should be read, an additional section *ssa* can be added to the approriate fixup file (in the case of *Multiline-Single* or *Single-Split* format) or by adding an *ssa* dictionary to the meta object in the specutils loader.

2.5 Data File Formats

2.5.1 All FITS files

FITS files can themselves contain significant amounts of metadata, via the header in each extension. Using the correct headers and including all the relevant metadata both makes it easier for Data Central to ingest and manage your survey data, and reduce the amount of issues that users of the survey will encounter. To this end, here are some things you can do to ensure that the FITS files are in the best condition:

1. astropy contains tools to ensure correctness of FITS files and metadata, run these tools on your FITS files before uploading them to Data Central. The ones we suggest running at a minimum are wcslist (for WCS) and verify() (for the FITS file itself).
2. Sanity check that the WCS metadata contained within the file: there may be keywords left over from the data reduction process (e.g. a 2d WCS for a 1d spectrum), or tools adding additional WCS keywords which override earlier values (e.g. `CRPIXjs` vs. `CDi_js` vs. `PCi_js`). astropy may be able to pick up some of these issues, but depending on how the data is stored, this is not always possible (e.g. a 2d WCS would not be detected for spectra stored in rows within a single HDU, whilst logically each row should be treated as being independent). A common cause of this is copying the contents of the FITS header from unreduced to a reduced version of the data, without checking the resulting header: make sure that you understand what each keyword in means, and that it is still applicable (e.g. 2dfdr included 2D WCS metadata, providing information about the wavelength-pixel mapping, and the fibre-pixel mapping, if you copy across the header when you extract a single spectra, the second WCS axis is logically invalid).
3. Specify all the units within the metadata, via standard keywords such as `BUNIT` and `CUNITn`. This means Data Central does not need to track units separately, and that standard tools (such as astropy) can display, convert

and manipulate data without users needing to manage units manually. See *Units within Data Central* for some unit-specific advice, which applies across all of the different data types and formats (including catalogues).

4. Specify bibliographic and provenance information in the primary header, via standard keywords such `ORIGIN` and `REFERENC` (please include as much as possible, even down to the specific data release the file is from). Whilst Data Central does not specifically use this information, tools such as specutils can use this to identify and load your files, and the files can understood outside the context of Data Central.
5. **Don't** include images/spectra/other data from other surveys within the same FITS file—this results in duplication of effort, increases the amount of code needed to manage these files (e.g. custom readers have to be written), and wastes storage and bandwidth. Instead, include references to other surveys via the survey object name (this is best done via catalogues, however this can additionally be stored in metadata within headers of the relevant files). The VO (Virtual Observatory) tools are designed for this kind of work, and future iterations of Data Central will expose more and more of these inter-survey links, and hence increase the richness of the visualisations.
6. Include metadata about the different HDUs within the FITS file. Use standard keywords (like `EXTNAME`) where possible, but there are other non-standard keywords such as `ROWn` or `ARRAYn` which have become unofficial conventions.

A reference containing all the standard FITS keywords can be found at https://heasarc.gsfc.nasa.gov/docs/fcg/standard_dict.html and further links to other sets of keywords and conventions can be found at https://fits.gsfc.nasa.gov/fits_standard.html.

2.5.2 Imaging

Coming soon.

2.5.3 IFS

cube_blue/cube_red

Coming soon.

2.5.4 Spectra

spectrum_1d

1D Spectral FITS files must have a single source spectrum in a single row. Flux, SN, sky emission can be included in different FITS extensions.

If you are registering a 1D spectra in your *<survey>_<datarelease>_product_meta.txt*, you must also include the following columns in the <survey>_<datarelease>_product_meta.txt file:

Column	Description
fluxHDU	Number of extensions containing source flux
snHDU	Number of extensions containing source signal to noise (can be blank if not provided)
skyHDU	Number of extensions containing sky emission (can be blank if not provided)
fluxUnitKeyword	FITS header keyword containing flux units (can be “ergs/sec/cm^2/ang”, “ergs/sec/cm^2/Hz”, ‘jansky”, ‘counts’)
fluxScaleKeyword	FITS header keyword containing scaling for flux units (numeric)
wavelengthUnitKeyword	FITS header keyword containing wavelength units (can be ‘ang’, ’nm’, ‘micron’, ‘m’)
CRPIXKeyword	FITS header keyword containing pixel reference point for wavelength scale.
CRVALKeyword	FITS header keyword containing pixel reference point value for wavelength scale.
CDELTKeyword	FITS header keyword containing pixel width for wavelength scale.

spectrum_2d

Coming soon.

2.6 Metadata File Formats

Metadata files are always pipe-delimited, and have the extension .txt

2.6.1 Documentation

If you would like formatted text to appear in the Schema Browser, you can supply a “documentation” entry for groups and tables. The file itself can contain html text.

Please use the following extensions (to denote the format of a particular documentation file):

.html contains html-formatted documentation

.txt contains plain text-formatted documentation

If you do not wish to supply documentation for a particular row, leave that entry blank.

If you would like to include email addresses in your documentation, please format as:

```
<a class="is-mailable" data-person="john.smith" data-domain="sydney.edu.au"></a>
```

This is an anti-spam measure: rather than placing raw html (easily harvested by spambots), a javascript-enabled browser is required to decode the address.

Documentation that you will want to update frequently should be maintained by the survey teams themselves in Data Central’s Documentation portal: Document Central. Contact us to get your team set up within the system if you have not done so already.

Important: Remember that the documentation used in the ingestion is the static, paper-like documentation, the documentation on Documentent Central is entirely separate.

2.7 Units within Data Central

Units and their representation is one of the more complex parts of the metadata Data Central tracks, due to the existence of differing, contradictory standards for representing units, and a large legacy of existing data which does not match a particular standard chosen. Data Central uses standard tools (such as astropy) to read, convert, write and generally handle units in a VO-compliant way as possible, and aim to fix all unit issues before any public data release. In general, Data Central will not modify existing data releases to fix issues with units.

In terms of actually representing units, we suggest reading VOUnits, the IVOA Recommendation on units within the Virtual Observatory, as it provides useful information about how existing unit standards (or conventions), both within astronomy and outside, specify how to encode units. The VOUnits document has nothing particularly unusual, though it does note some corner cases worth taking note of, such as treating the string `'Pa'` as Pascals rather than petayears (which would happen if naively if `'P'` was always treated as representing Peta).

In general, there is nothing particularly strange you need to do to produce data with VO-compliant units, but there are two things you should probably check: the capitalisation and pluralisation of the units; and encoding dimensionless or unitless quantities.

2.7.1 Capitalisation and Pluralisation of Units

Astropy by default reads only the singular form of a unit (e.g. “second” over “seconds” or “Angstrom” over “Angstroms”), as generally this is the preferred form in most unit standards. Data Central handles plural forms for existing data releases, but for maximum compatibility, the singular form should be used for new data (we are unaware of any software which rejects the singular form).

Similarly, lower case (except for units named after a person) is to be used, with capitalisation of the first letter if the unit is named after a person (e.g. “Angstrom” over “angstrom”). All caps (every letter being upper case) will throw errors in astropy, and therefore should be avoided in new data.

2.7.2 Encoding Dimensionless or Unitless Quantities

Neither the FITS standard nor the IAU Style Guide discuss the representation of dimensionless/unitless quantities, so there is no specific representation in standard FITS. There are some differing conventions mentioned in the VOUnits document, with the VOUnits document recommending using an empty string.

For normalised quantities, such as normalised intensity, we suggest using the empty string for the unit, and describe the quantity in the comment (e.g. have the FITS comment for the field be “Normalised Intensity” for something stored as normalised intensity).

CHAPTER

THREE

PROCESSES

If you have any questions about this documentation during the preparation of your data release, please contact us.

3.1 Response Timescales

We endeavour to implement any change-requests as soon as possible, however the DC team is currently relatively small. Please see below for a rough estimate on update timescales. Please note, these are a guide only, subject to the complexity of the update requested, and additional constraints on our time (e.g., other survey-team data releases, conferences, funding milestones etc).

3.1.1 Data and Schema Changes

For critical changes (e.g., data product files/catalogues need to be replaced with new versions), we will work towards an update turnaround of ~1 week. For non-critical changes (e.g., minor schema changes) we guarantee update timescales less than 1 month. To clarify whether a change is critical or non-critical, contact us.

3.1.2 Interface Issue

For critical interface issues (e.g., blocking data access) we aim to address any issue submitted by any user within 1 week (though usually within 48 hours). For non-critical issues (e.g., mis-alignment on mobile view) we aim to publish a fix within 1 month.

3.1.3 Feature Requests

Data Central is developed with its users in mind. We'd love to hear your suggestions for new features you'd like to see developed, plus give you the opportunity to vote on the functionality you'd like to see prioritized.

Please suggest new features and vote on existing suggestions on our requests page. We use this information to guide our milestones and future development effort.

CHAPTER

FOUR

POLICIES

We reserve the right at any time to remove data that is in violation of any data usage licenses or agreements.

4.1 Data

When you release data through Data Central, all data and meta data are available for public download and distribution. Unless you request otherwise, data will be publicly released with a Creative Commons Attribution 4.0 International (also known as CC BY 4.0). See https://creativecommons.org/licenses/by/4.0/ for more details.

Who owns the data? You do! Any data that is uploaded to Data Central belongs to the researchers who created it. We are merely custodians of your data, allowing you to share it publicly or with your team.

What data can we share? We will only share public data and only with the owner's permission.

With whom can we share data? If data has been made public, then anyone has access to it. Private team data is only available to team members, and access is in the control of the team.

4.2 Maintenance

We aim to push new code to the production server on a regular basis and as such, guarantee a quick turnaround for bug fixes and data updates. Downtime and maintenance windows are currently unscheduled, but we aim to keep disruption at a minimum (i.e., <1h). You can check our twitter for maintenance notices @DataCentralOrg.

CHAPTER FIVE

LICENSE

Copyright 2019 Data Central

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER SIX

SUPPORT

Got a question? You can email us at contact@datacentral.org.au or send us a message via our support site.

INDEX