How to Load a Dataset with load_data_proj

This content was copied to the internal website:
https://internal.eol.ucar.edu/content/load-dataset-loaddataproj
On the internal website the images have disappeared. Copy them there from here?

What is it for?

`load_data_proj` is a python based GUI tool that simplifies the dataset loading process. This readme will go through the steps that are needed to publish a dataset onto the production server.

To run `load_data_proj`, first change directories to:

```
     cd /net/work/bin/scripts/insert/loaddata
```

Then run the `load_data_proj` script by:

```
     ./load_data_proj
```

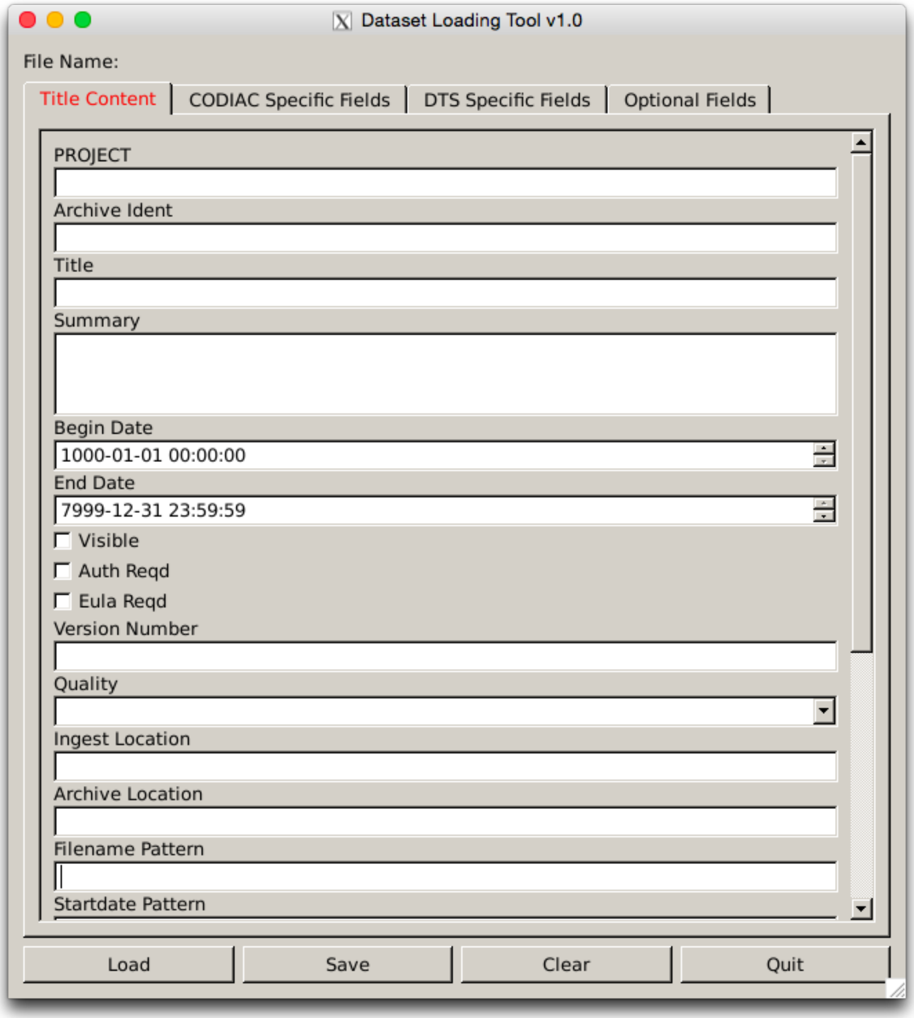
The metadata needed to load a dataset are broken up into four different tabbed sections.

Most fields have specific requirements. To view the requirements, simply hover over the field (or one can view the field messages yaml file:

```
     cd /net/work/bin/scripts/insert/loaddata/config
     cat fieldMsgs.yml
```

)

Title Content Tab



Begin Date: 1000-01-01 00:00:00

End Date: 7999-12-31 23:59:59

□ Visible

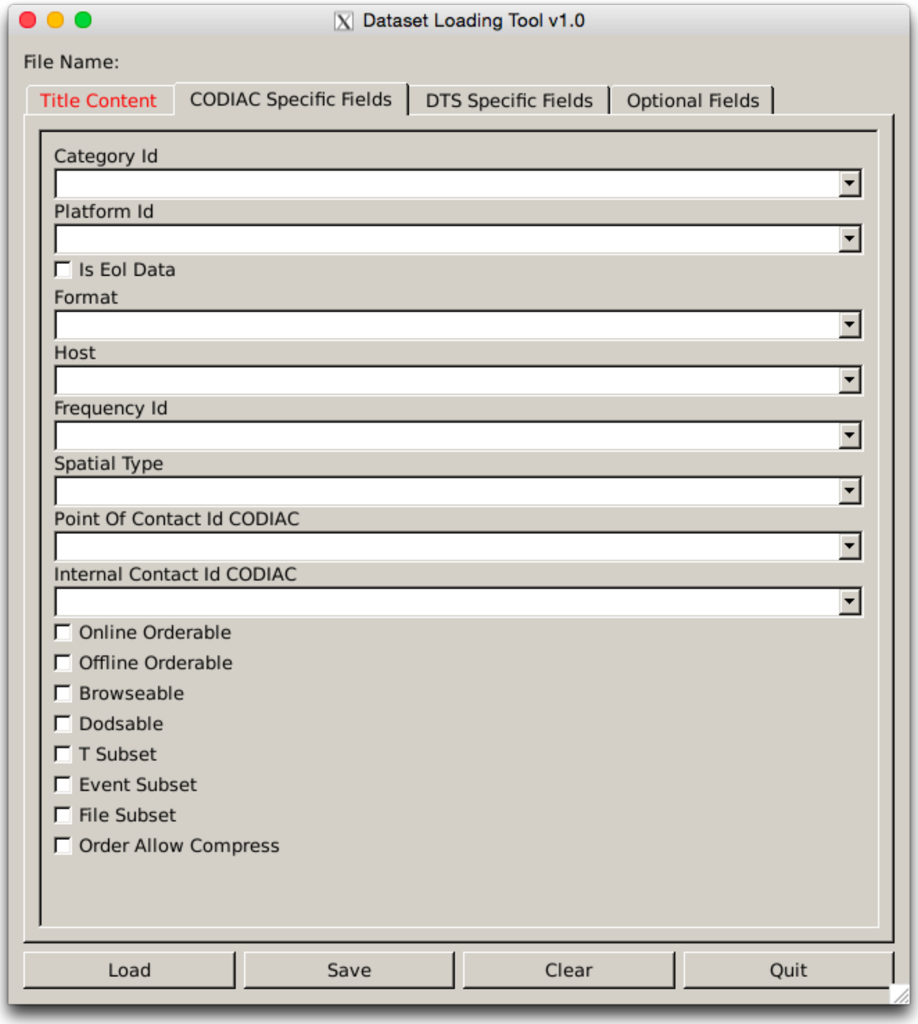
□ Auth Reqd

□ Eula Reqd

Version Number:

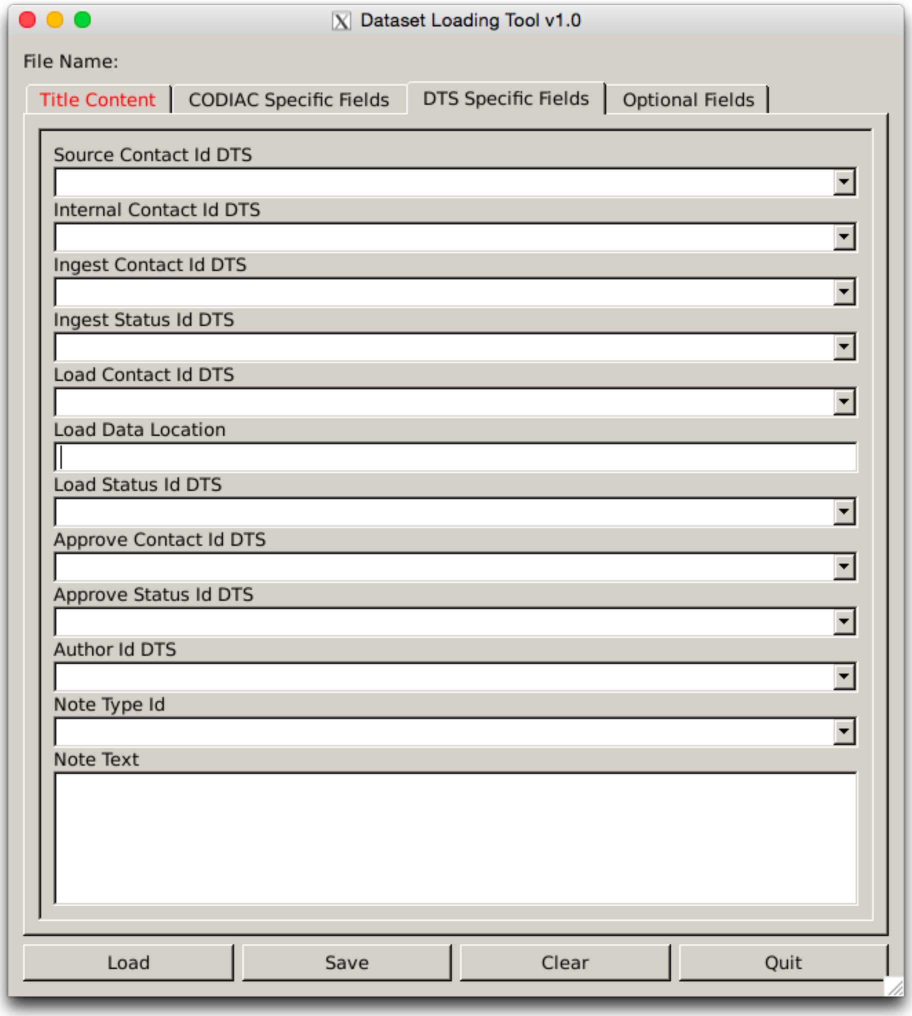
The title content tab contains required fields that are dataset based therefore need to be changed for every dataset rather than every project.

Codiac Specific Fields



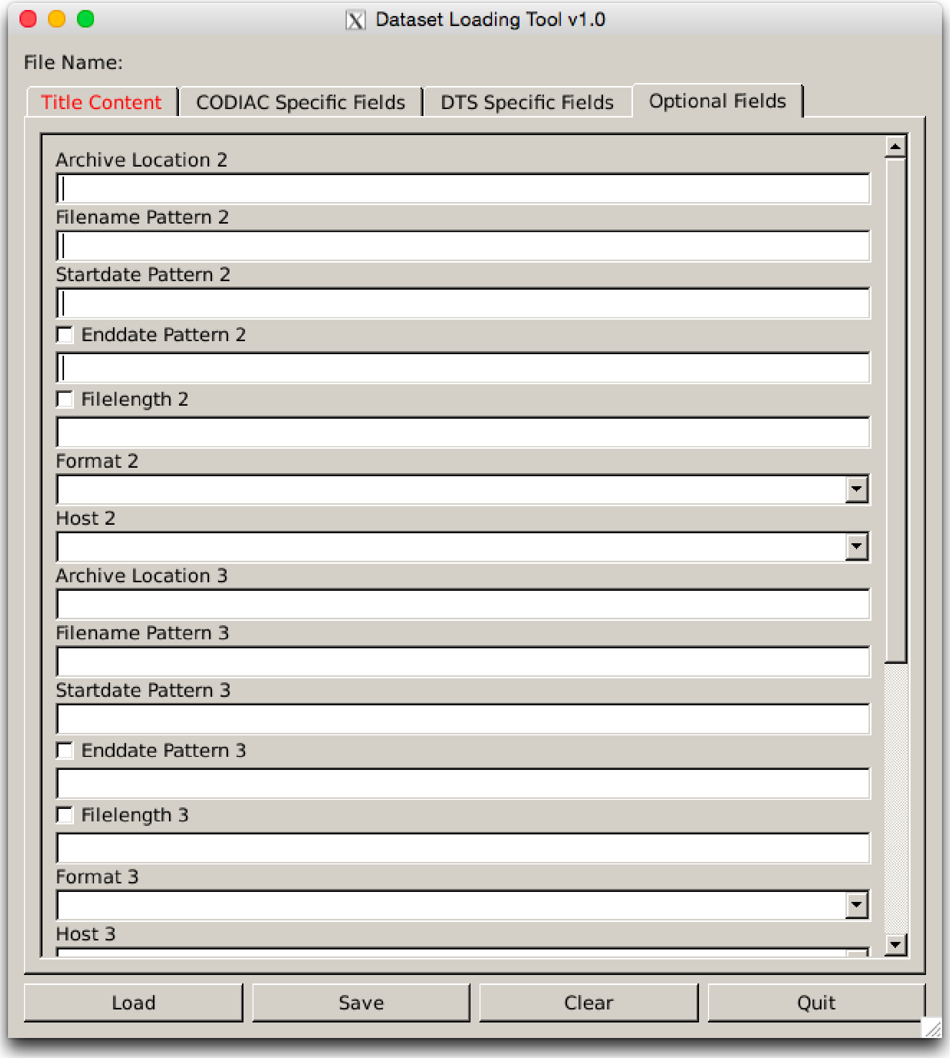
The CODIAC Specific Fields tab contains additional required CODIAC fields that are data-type specific so should remain constant for a given data type from project to project.
However, always make sure that all entries are correct before deploying to the production server.

DTS Specific Fields



The DTS Specific Fields tab contains DTS specific required fields that are data-type specific so should remain constant for a given data type from project to project.
However, always make sure that all entries are correct before deploying to the production server.

Optional Fields

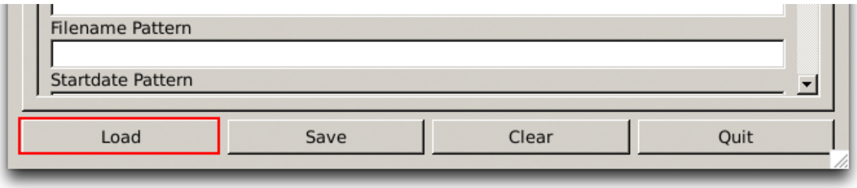


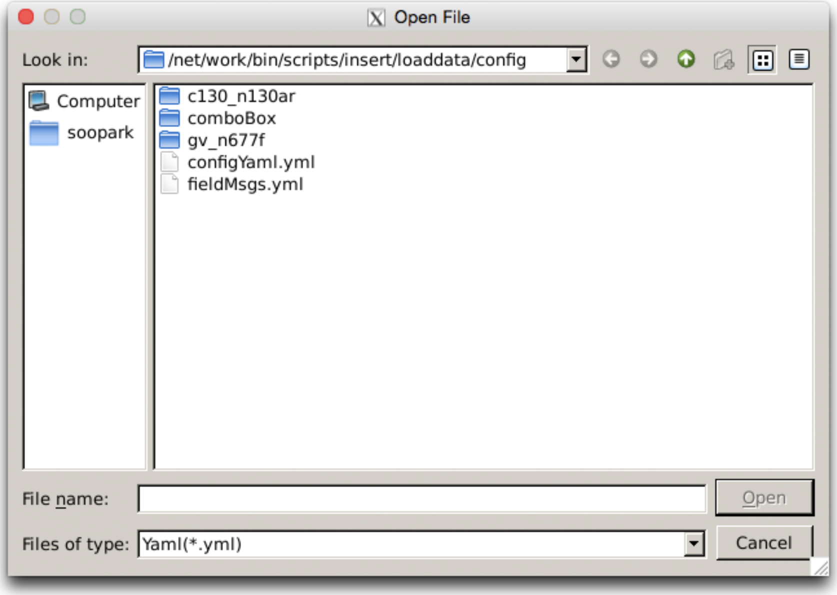
The tab contains optional fields:

- to add a second file type (fills the role of a second set of brackets in a cfg-file)
- to add contacts (EOL and the internal contact are already added)

These fields may be left blank, but the other fields in the other tabs may not.

Loading YAML files





Look in: /net/work/bin/scripts/insert/loaddata/config

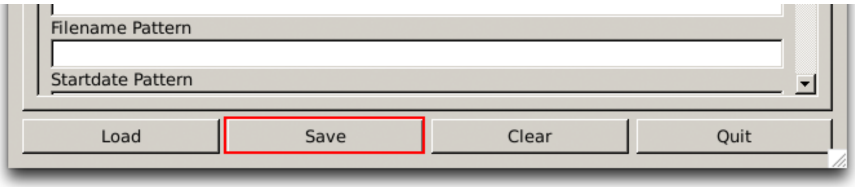
Files of type: Yaml(*.yml)

There are two ways of loading a yaml file:

- load a configuration yaml file for a specific data type from the `config` directory.
- load a previously created project yaml file from the `cfg-files` directory.

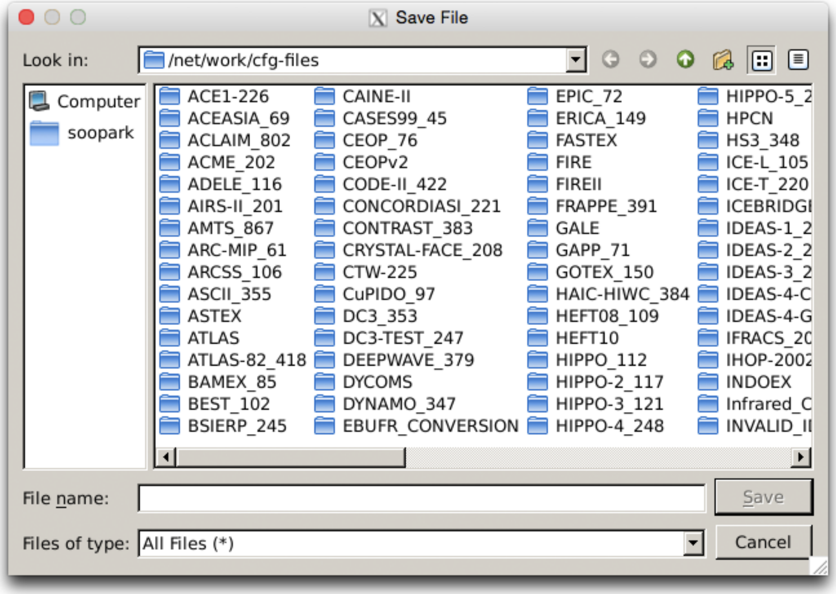
Load a config yaml file to create a new dataset yaml file.
For example, the figure above shows a loaded config yaml file for c130 downward looking camera.
The purpose of a config yaml file is to have a template for creating datasets that occur repeatedly from project to project.

Saving Yaml files



Filename Pattern

Startdate Pattern



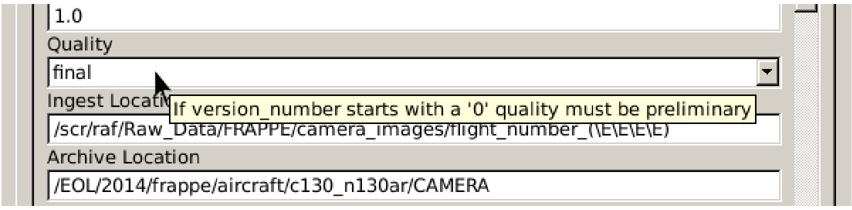
Look in: /net/work/cfg-files

ACE1-226
ACEASIA_69
ACLAIM_802
ACME_202
ADELE_116
AIRS-II_201
AMTS_867
ARC-MIP_61
ARCSS_106
ASCII_355
ASTEX
ATLAS
ATLAS-82_418
BAMEX_85
BEST_102
BSIERP_245
CAINE-II
CASES99_45
CEOP_76
CEOPv2
CODE-II_422
CONCORDIASI_221
CONTRAST_383
CRYSTAL-FACE_208
CTW-225
CuPIDO_97
DC3_353
DC3-TEST_247
DEEPWAVE_379
DYCOMS
DYNAMO_347
EBUFR_CONVERSION
EPIC_72
ERICA_149
FASTEX
FIRE
FIREII
FRAPPE_391
GALE
GAPP_71
GOTEX_150
HAIC-HIWC_384
HEFT08_109
HEFT10
HIPPO_112
HIPPO-2_117
HIPPO-3_121
HIPPO-4_248
HIPPO-5_2
HPCN
HS3_348
ICE-L_105
ICE-T_220
ICEBRIDGI
IDEAS-1_2
IDEAS-2_2
IDEAS-3_2
IDEAS-4-C
IDEAS-4-G
IFRACS_2C
IHOP-2002
INDOEX
Infrared_C
INVALID_II

File name:

Files of type: All Files (*)

In order to successfully save a yaml file, all fields except unfilled optional fields must pass the save check.



1.0

Quality

final

Ingest Location

/scr/raf/Raw_Data/FRAPPE/camera_images/flight_number_(\E\E\E\E)

If version_number starts with a '0' quality must be preliminary

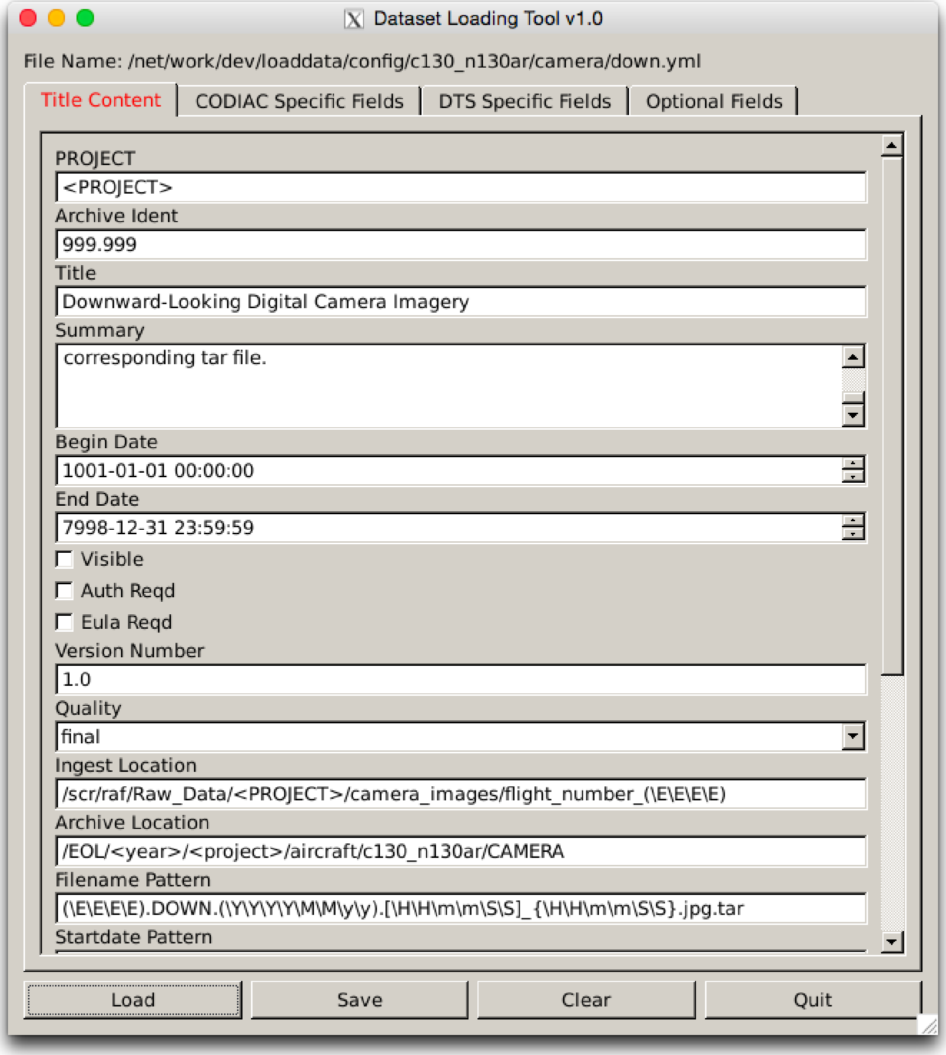
Archive Location

/EOL/2014/frappe/aircraft/c130_n130ar/CAMERA

For details of the check requirements, reference the tips available when hovering over a field.

Save the complete yaml file in the respective projects directory.

Example: FRAPPE Downward Camera



File Name: /net/work/dev/loaddata/config/c130_n130ar/camera/down.yml

PROJECT
<PROJECT>

Archive Ident
999.999

Title
Downward-Looking Digital Camera Imagery

Summary
corresponding tar file.

Begin Date
1001-01-01 00:00:00

End Date
7998-12-31 23:59:59

□ Visible

□ Auth Reqd

□ Eula Reqd

Version Number
1.0

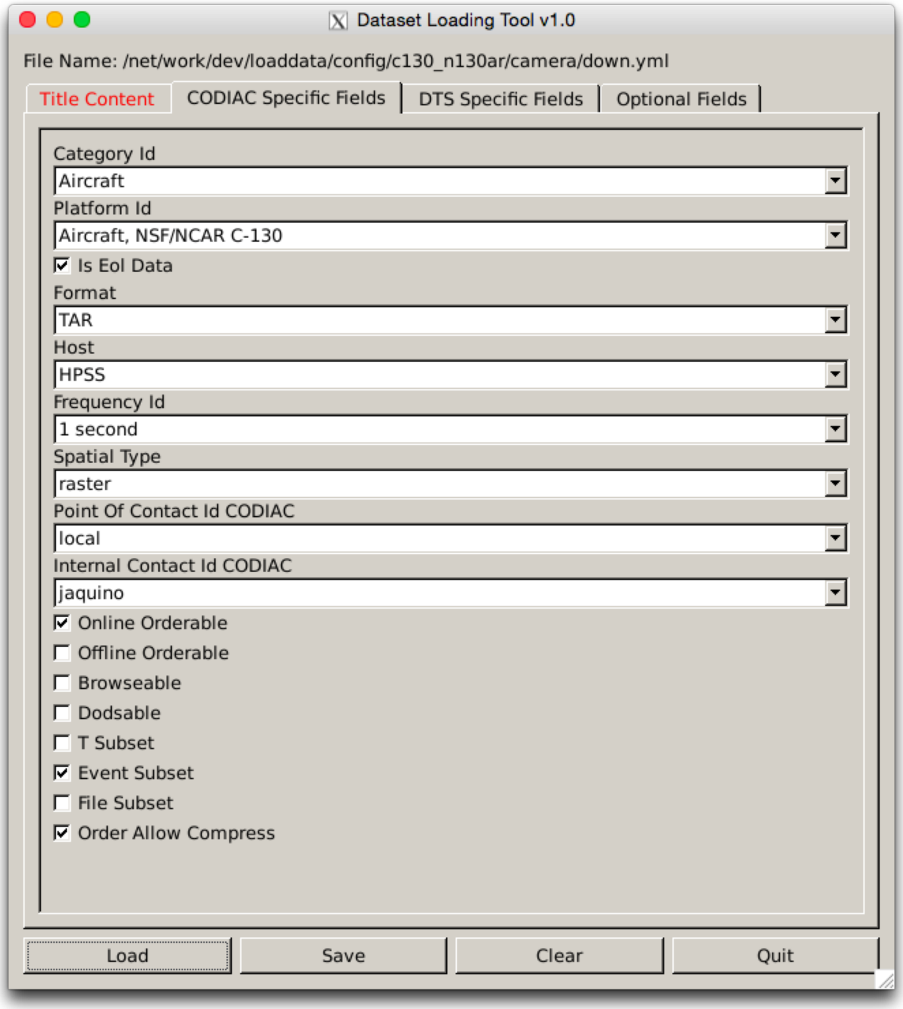
Quality
final

Ingest Location
/scr/raf/Raw_Data/<PROJECT>/camera_images/flight_number_(\E\E\E\E)

Archive Location
/EOL/<year>/<project>/aircraft/c130_n130ar/CAMERA

Filename Pattern
(\E\E\E\E).DOWN.(\Y\Y\Y\Y\M\M\y\y).[\H\H\m\m\S\S]_{\H\H\m\m\S\S}.jpg.tar

Startdate Pattern



File Name: /net/work/dev/loaddata/config/c130_n130ar/camera/down.yml

Category Id
Aircraft

Platform Id
Aircraft, NSF/NCAR C-130

☑ Is Eol Data

Format
TAR

Host
HPSS

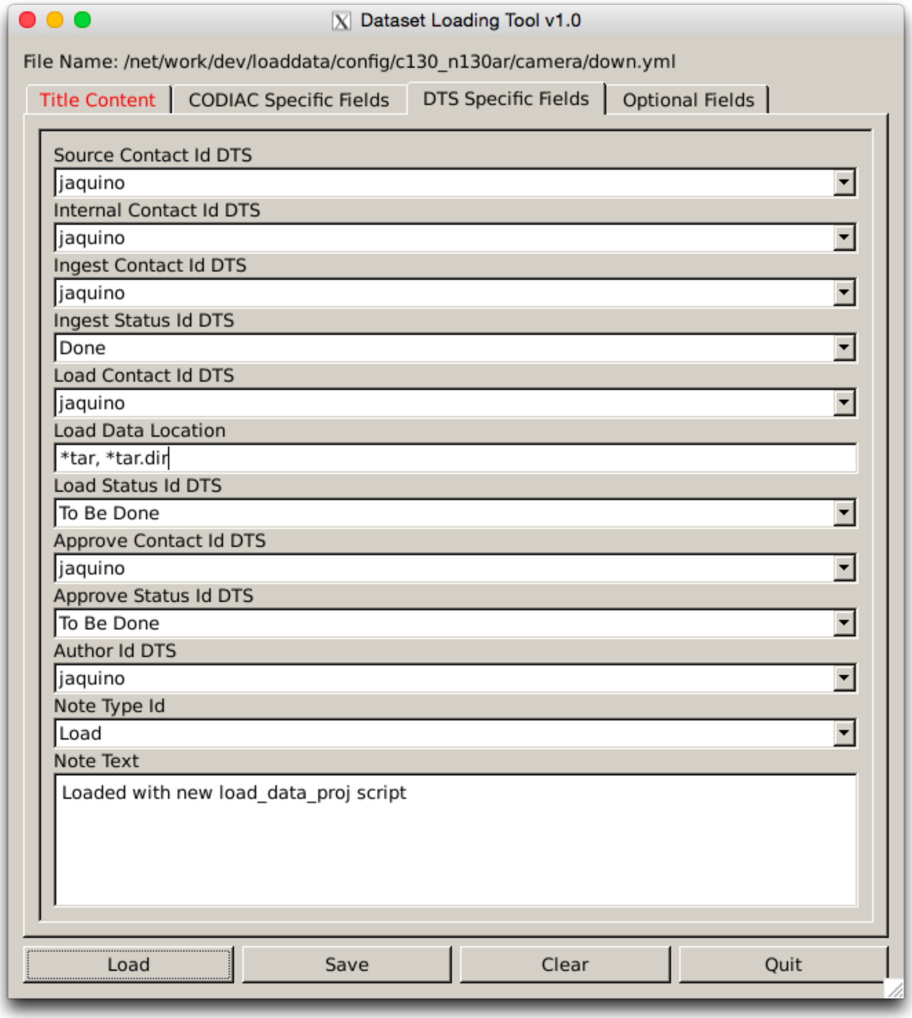
Frequency Id
1 second

Spatial Type
raster

Point Of Contact Id CODIAC
local

Internal Contact Id CODIAC
jaquino

☑ Online Orderable
☐ Offline Orderable
☐ Browseable
☐ Dodsable
☐ T Subset
☑ Event Subset
☐ File Subset
☑ Order Allow Compress



File Name: /net/work/dev/loaddata/config/c130_n130ar/camera/down.yml

Source Contact Id DTS
jaquino

Internal Contact Id DTS
jaquino

Ingest Contact Id DTS
jaquino

Ingest Status Id DTS
Done

Load Contact Id DTS
jaquino

Load Data Location
*tar, *tar.dir

Load Status Id DTS
To Be Done

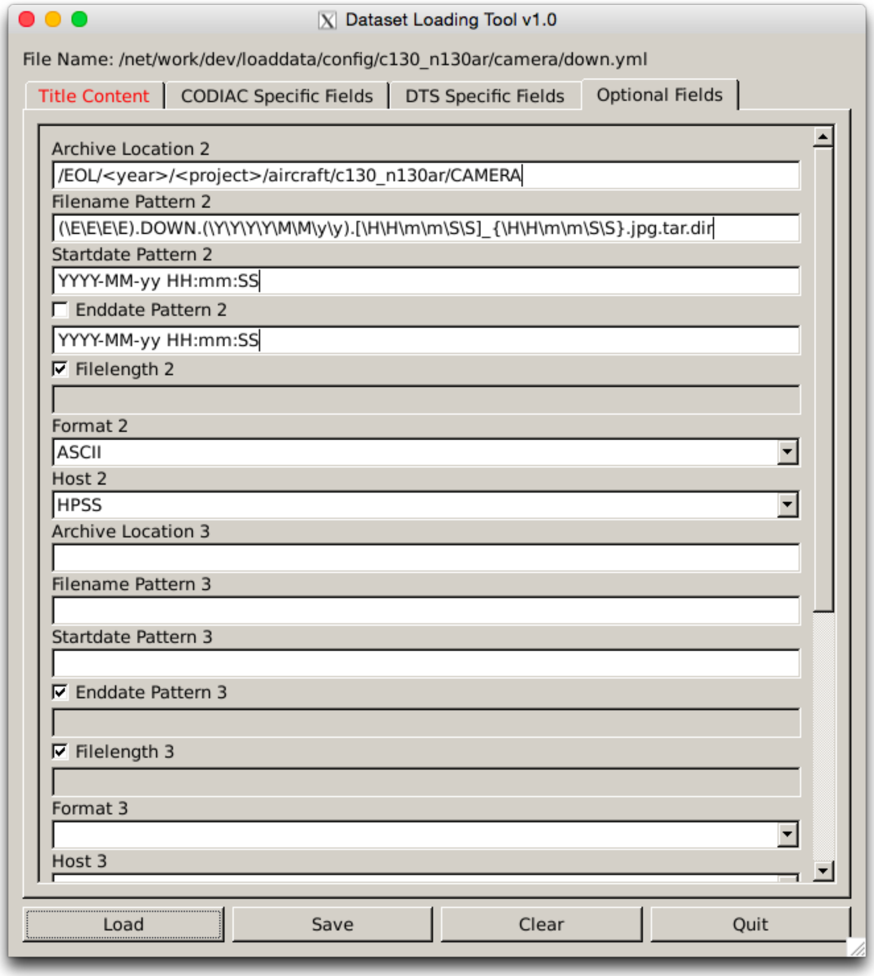
Approve Contact Id DTS
jaquino

Approve Status Id DTS
To Be Done

Author Id DTS
jaquino

Note Type Id
Load

Note Text
Loaded with new load_data_proj script



File Name: /net/work/dev/loaddata/config/c130_n130ar/camera/down.yml

Archive Location 2

/EOL/<year>/<project>/aircraft/c130_n130ar/CAMERA

Filename Pattern 2

`(\E\E\E\E).DOWN.(\Y\Y\Y\Y\M\M\y\y).[\H\H\m\m\S\S]_{\H\H\m\m\S\S}.jpg.tar.dir`

Startdate Pattern 2

YYYY-MM-yy HH:mm:SS

□ Enddate Pattern 2

YYYY-MM-yy HH:mm:SS

☑ Filelength 2

Format 2

ASCII

Host 2

HPSS

Archive Location 3

Filename Pattern 3

Startdate Pattern 3

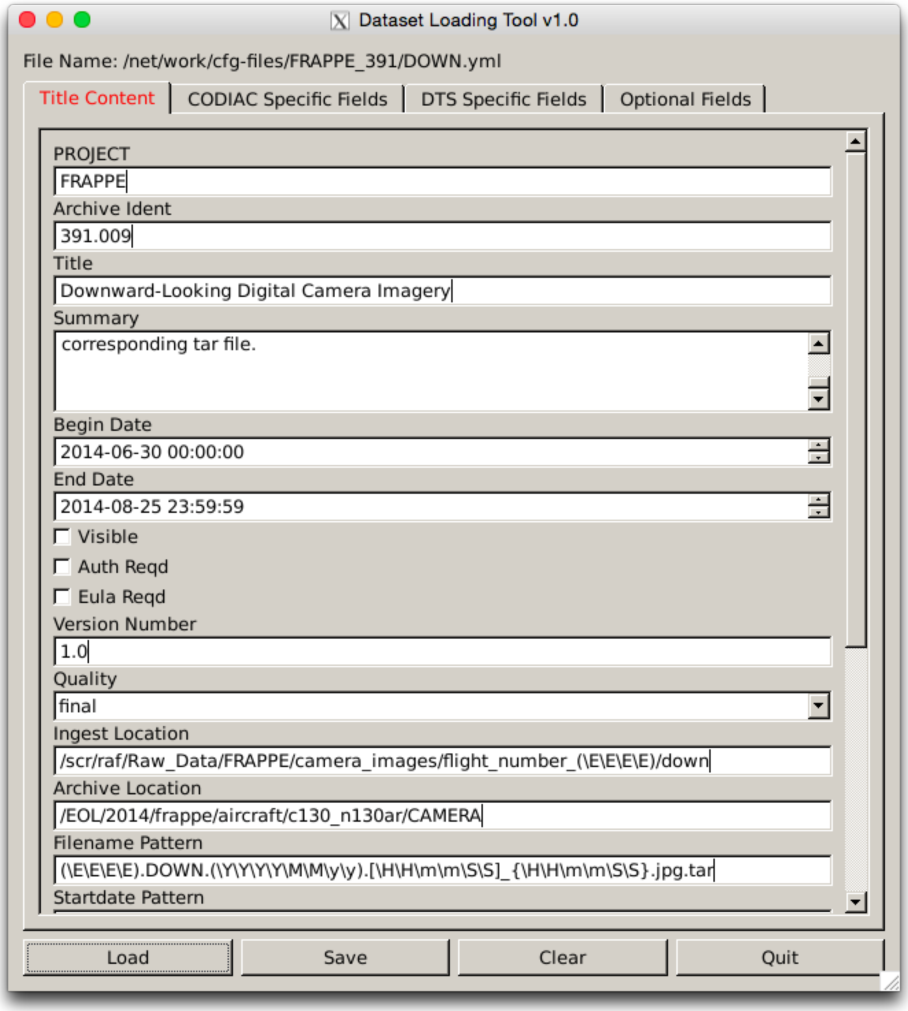
☑ Enddate Pattern 3

☑ Filelength 3

Format 3

Host 3

1. Replace ALL **<*>** with their corresponding values. For example, **<PROJECT>** is replace by FRAPPE, **<project>** with frappe, and **<year>** with 2014.
2. Replace the Archive identifier with the desired dataset ID.
3. Replace the begin and End Date with the begin and end dates indicated in the datafile names.
4. If the dataset is final, the version number must begin with a non ‘0’ number. ex) 1.0, 5.0
5. The checkboxes next to Enddate Pattern and Filelength indicate their disabled status. Note that for this particular dataset, filelength has been disabled and enddate is being used. For a successful save, either one of the two fields MUST be disabled and the other must be used in in its place.
6. Follow standard cfg- formatting for patterns.



Dataset Loading Tool v1.0

File Name: /net/work/cfg-files/FRAPPE_391/DOWN.yml

PROJECT
FRAPPE

Archive Ident
391.009

Title
Downward-Looking Digital Camera Imagery

Summary
corresponding tar file.

Begin Date
2014-06-30 00:00:00

End Date
2014-08-25 23:59:59

□ Visible

□ Auth Reqd

□ Eula Reqd

Version Number
1.0

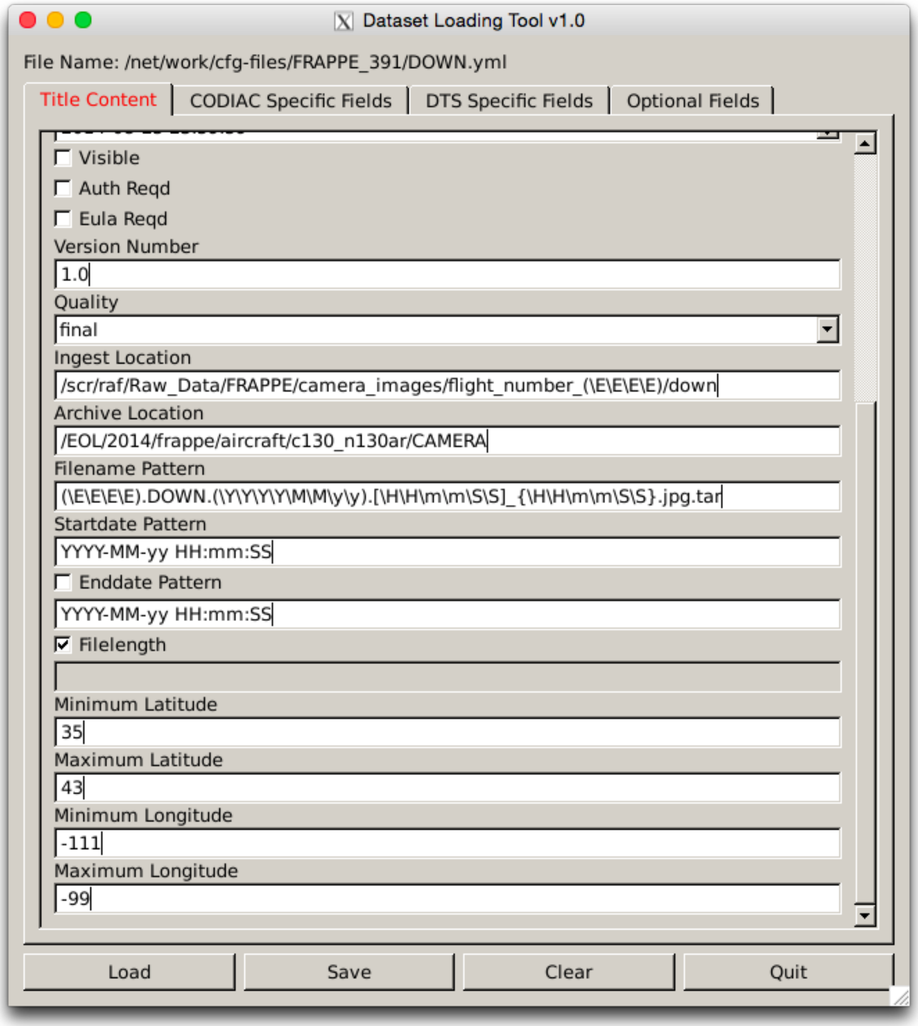
Quality
final

Ingest Location
/scr/raf/Raw_Data/FRAPPE/camera_images/flight_number_(\E\E\E\E)/down

Archive Location
/EOL/2014/frappe/aircraft/c130_n130ar/CAMERA

Filename Pattern
(\E\E\E\E).DOWN.(\Y\Y\Y\Y\M\M\y\y).[\H\H\m\m\S\S]_{\H\H\m\m\S\S}.jpg.tar

Startdate Pattern



File Name: /net/work/cfg-files/FRAPPE_391/DOWN.yml

□ Visible
□ Auth Reqd
□ Eula Reqd

Version Number: 1.0

Quality: final

Ingest Location: /scr/raf/Raw_Data/FRAPPE/camera_images/flight_number_(\E\E\E\E)/down

Archive Location: /EOL/2014/frappe/aircraft/c130_n130ar/CAMERA

Filename Pattern: (\E\E\E\E).DOWN.(\Y\Y\Y\Y\M\M\y\y).[\H\H\m\m\S\S]_{\H\H\m\m\S\S}.jpg.tar

Startdate Pattern: YYYY-MM-yy HH:mm:SS

□ Enddate Pattern: YYYY-MM-yy HH:mm:SS

☑ Filelength

Minimum Latitude: 35

Maximum Latitude: 43

Minimum Longitude: -111

Maximum Longitude: -99

Adding New Fields

Adding new fields is very simple. First, change directories to:

```
cd /net/work/bin/scripts/insert/loaddata/config
```

Then

```
vi configYaml.yml
```

Here are the stpes to adding a new field (You can find these instructions in the configYaml file as well):

1) Decide the placement of your new field. All fields with the "titleSep" will automatically act a s a tab separator for all the fields and create a new tab for all fields under the titleSep up until the new titleSep or the end of the file.
2) Decide the name of your new field:
   a) lower-case every character
   b) put an underscore ("_") where you want a space
   c) all the first letters will be capitalized. ex) if you want the label of your dataset to be "Archive Ident" in the GUI, you must enter it as "archive_ident" in the config file.
3) Decide the type of your new field. There are five types of fields that are currently available:

a) lineEdit: fields that are one line long
b) plainTextEdit: fields that are multiple lines long
c) comboBox: drop down list with additional config file required
d) checkBox: check box field
e) dateTimeEdit: date and time field
f) titleSep: tab separator. If you'd like a new tab, please discuss with Soo first.

4) decide on the requirements of your new field. (ex. it must be in decimals, or it cannot be more than certain characters/words long, the date must be in between a certain range...)
5) Depending on the requirements, you may have to write additional lines of code to implement the new checking method. If your checking method is already implemented in the field modules, you do not have to write any additional code and will only be checked for completion. The checking method will have to go in two different files:
   a) the field type module file. For example, if the filed type is lineEdit, this new checking method will go in lineEdit.py Place the new function at the bottom of the code.
   b) load_data_proj file under errorMsgs function.
6) Once the checking method implementation is complete, add your new field with the type in the exact format of currently existing fields.

If you aren't sure of how to write a new checking method, please contact Soo or Janine for assistance. Also, if you have added a new field, please still contact Soo or Janine for verification.

Adding New ComboBox Fields

Adding new comboBox fields require an extra bit of work. comboBox fields require their own config files to populate the drop down list, so before adding the field to the configYaml file, you must create a config file for the respective field and save it to the

`/net/work/bin/scripts/insert/loaddata/config/comboBox`

directory.

The comboBox config file name and the field name in the yaml dataset must match EXACTLY. Or els the script will NOT work.

For example, if the name of the comboBox field is "Platform Id", the entry in the configYaml must be "`platform_id: comboBox`" and there must be a config file named "`platform_id.yml`" in the above directory with the content:

```
platform_id:
  - 2: aircraft
  ...
```

load_a_dataset

Once you have saved a yaml file for your dataset, the load_a_dataset script is used to: create a new dataset n codiac, load the files to the new dataset, create an entry in the DTS for this dataset, and print a reminder of tasks that need to be done to complete the dataset loading.

It is comprised of three different parts:

- `insert_proj_dataset` (USAGE: `../insert_proj_dataset add_dataset=down.yml`)

    - This script is used to insert projects and datasets into zith9 with the help of setup files to facilitate automated data loading.
- `insert_dts` (USAGE: `../insert_dts add_dts=down.yml`)
    - This script inserts datasets into dmg_dts with the help of setup files to facilitate automated data loading.
- `insert_multiple_files` (USAGE: `../insert_multiple_files down.yml`)
    - This script inserts multiple datafiles at once into the database for a dataset.

(These steps can be run at a time - useful if the script fails part way through.)

To run load_a_dataset, change directories into where the dataset yaml file is residing in (most likely the cfg-files directory) and run this line of code:

```
/net/work/bin/scripts/insert/loaddata/load_a_dataset <dataset>.yml
```

The code will run the three sub scripts in order: `insert_proj_dataset, insert_dts, insert_multiple_files`. The user will be prompted at the end of each script to confirm the database entries before they are submitted.
The code will print a set of helpful reminder as to what else may need to be done to finalize the datasets.

Email Janine and Soo with questions, comments, suggestions, bugs, etc.
Janine: jaquino@ucar.edu
Soo: soopark@ucar.edu

1. Decide the placement of your new field. All fields with the "titleSep" will automatically act as a tab separator for all the fields and create a new tab for all fields under the titleSep until the new titleSep or the end of the file.
2. Decide the name of your new field:
    a. lower-case every character
    b. put an underscore ("_") where you want a space
    c. all the first letters will be capitalized. ex) if you want the label of your dataset to be "Archive ident" in the GUI, you must enter it as "archive_ident" in the config gile.
3. Decide the type of your new field. There are five types of fields that are available:
    a. lineEdit: fields that are one line long
    b. plainTextEdit: fields that are multiple lines long
    c. comboBox: drop down list with additional config file required. Please read "Adding New ComboBox Fields" for more details.
    d. checkBox: check box field
    e. dateTimeEdit: date and time field
    f. (!!) titleSep: tab separator. if you'd like a new tab, please talk to Soo.
4. Decide on the requirements of your new field. (ex) it must be in decimals, cannot be more than certain characters/words long... etc.)
5. Depending on the requirements, you may have to write additional lines of code to implement the new checking method. If your checking method is already implemented, you do not have to write any additional code. The checking method will have to go in two different fiels:

   a. the field type module file. For example, if the field type is linEdit, this new checking method will go in lineEdit.py. Place the new function at the bottom of the code.
   b. load_data_proj file under errorMsgs function.

6. Once the checking method is complete, add your new field with the type in the exact format of currentyl existing fields.