



**STScI** | SPACE TELESCOPE  
SCIENCE INSTITUTE

EXPANDING THE FRONTIERS OF SPACE  
ASTRONOMY

# The Hubble Space Telescope (HST) Advanced Camera for Surveys (ACS) Quicklook Application

---

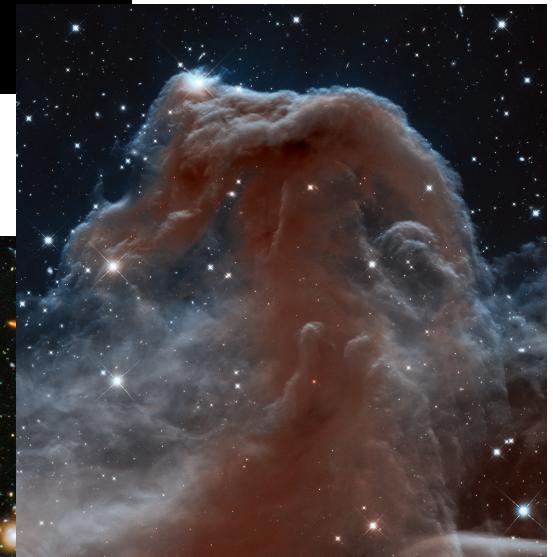
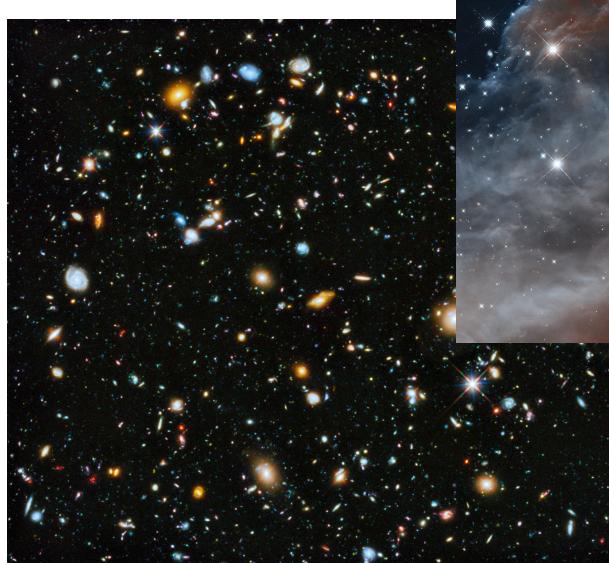
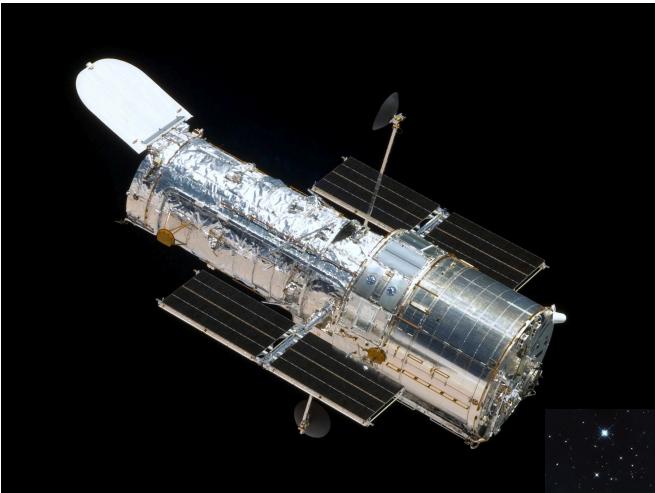
Matthew Bourque

October 19, 2017



# The Hubble Space Telescope (HST)

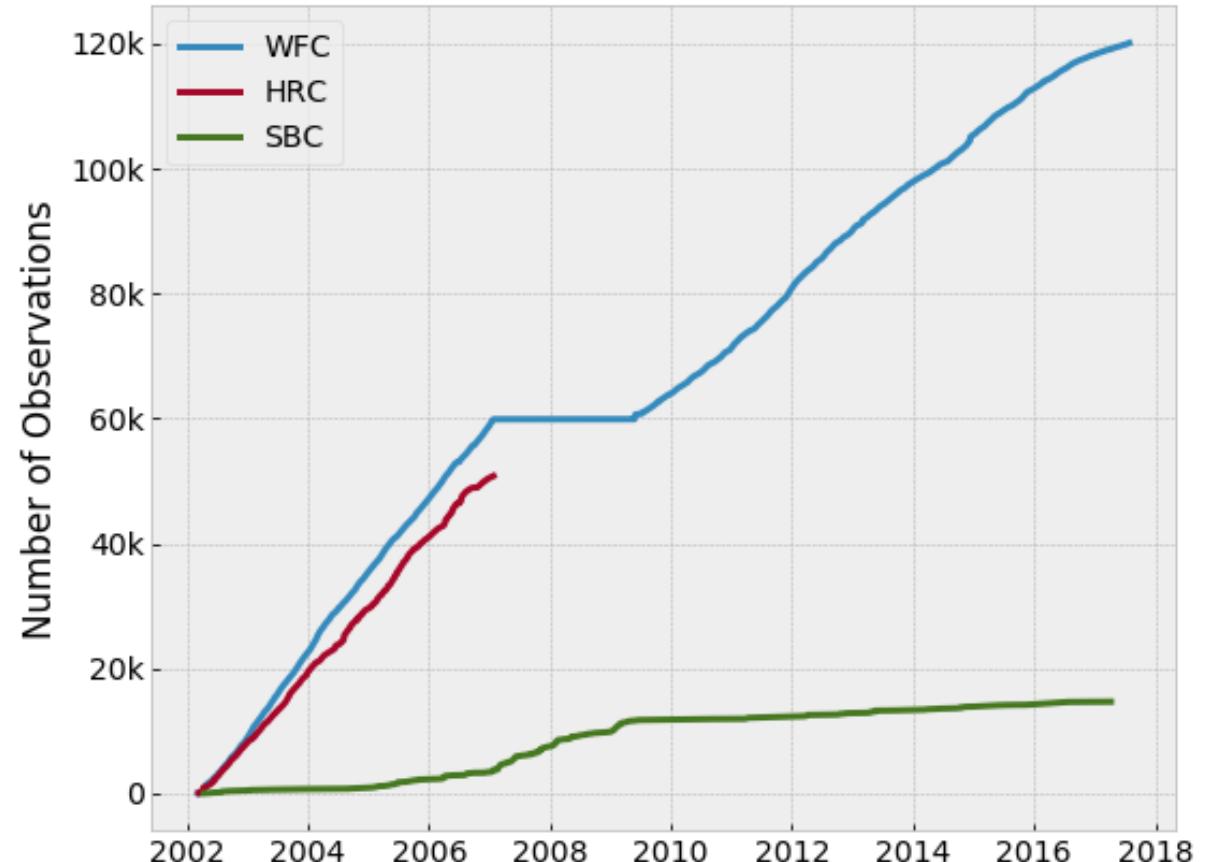
- HST was launched in 1990
- Over the years it has housed 12 instruments
- 5 Servicing Missions
- Currently there are 4 instruments on board:
  - Cosmic Origins Spectrograph (COS) (2009)
  - Space Telescope Imaging Spectrograph (STIS) (1990)
  - Wide Field Camera 3 (WFC3) (2009)
  - Advanced Camera for Surveys (ACS) (2002)





## The Advanced Camera for Surveys (ACS)

- Comprised of three detectors:
  - Wide Field Channel (WFC)
  - High Resolution Channel (HRC)
  - Solar Blind Channel (SBC)
- ACS experienced an electronics failure in 2006, temporarily disabling the WFC detector and permanently disabling the HRC detector.





# The HST/ACS Quicklook Application (acsq1)

- Database-driven web application for discovering and viewing HST/ACS images and corresponding metadata
- All publicly-available on-orbit data
- Comprised of four major components:
  - NFS Filesystem
  - MySQL Database
  - Python/Flask-based Web Application
  - Software Package
- Inspired by a similar application for the HST/WFC3 instrument (“WFC3 Quicklook”)

The screenshot shows a web browser window titled "ACS Quicklook" with three tabs. The central tab displays a grayscale astronomical image of a star-forming region with dense clouds of interstellar gas and dust. Above the image, the proposal identifier "jcs718kmq" is shown, along with the title "Proposal 14039: Broad-band imaging of Westerlund 2". Below the title, it says "Image 40 of 48" and "No FITS downloads available yet". To the right of the image, there is a column of metadata: "Expstart: 56991.84858445", "Filter: CLEAR1L/F814W", "Aperture: WFCENTER", "Exposure time: 348.0", "Expflag: NORMAL, Quality: None", "Coordinates: 156.02387, -57.76315", "Target: WESTERLUND-2", and "PI: Zolt Levay". At the bottom of the page, the text "Space Telescope Science Institute" is visible.



# Data Structure: FITS Files

FITS = Flexible Image Transport System

Extension	Purpose	Image Dimensions (pixels)	Data Type
0	Primary header	–	String
1	SCI, Chip 2	(4096, 2048)	Float
2	ERR, Chip 2	(4096, 2048)	Float
3	DQ, Chip 2	(4096, 2048)	Integer
4	SCI, Chip 1	(4096, 2048)	Float
5	ERR, Chip 1	(4096, 2048)	Float
6	DQ, Chip 1	(4096, 2048)	Integer

ACS/WFC FITS file structure

Extension	Purpose	Image Dimensions (pixels)	Data Type
0	Primary header	–	String
1	SCI	(1024, 1024)	Float
2	ERR	(1024, 1024)	Float
3	DQ	(1024, 1024)	Integer

ACS/WFC and ACS/SBC FITS file structure



# Data Structure: FITS file headers and key metadata

```
SIMPLE = T / data conform to FITS standard
BITPIX = 16 / bits per data value
NAXIS = 0 / number of data axes
EXTEND = T / File may contain standard extensions
NEXTEND = 6 / Number of standard extensions
GROUPS = F / image is in group format
DATE = '2016-09-22' / date this file was written (yyyy-mm-dd)
FILENAME= 'j6mf16lhq_raw.fits' / name of file
FILETYPE= 'SCI' / type of data found in data file

TELESCOP= 'HST' / telescope used to acquire data
INSTRUME= 'ACS' / identifier for instrument used to acquire data
EQUINOX = 2000.0 / equinox of celestial coord. system

/ DATA DESCRIPTION KEYWORDS

ROOTNAME= 'j6mf16lhq' / rootname of the observation set
IMAGETYP= 'DARK' / type of exposure identifier
PRIMESI = 'ACS' / instrument designated as prime

/ TARGET INFORMATION

TARGNAME= 'DARK' / proposer's target name
RA_TARG = 0.0000000000E+00 / right ascension of the target (deg) (J2000)
DEC_TARG= 0.0000000000E+00 / declination of the target (deg) (J2000)

/ PROPOSAL INFORMATION

PROPOSID= 9433 / PEP proposal identifier
LINENUM = '16.055' / proposal logsheet line number
PR_INV_L= 'Bernstein' / last name of principal investigator
PR_INV_F= 'Gary' / first name of principal investigator
PR_INV_M= '' / middle name / initial of principal investigator

/ EXPOSURE INFORMATION

SUNANGLE= 93.563698 / angle between sun and V1 axis
MOONANGL= 33.222004 / angle between moon and V1 axis
SUN_ALT = 68.062172 / altitude of the sun above Earth's limb
FGSLOCK = 'FINE' / commanded FGS lock (FINE,COARSE,GYROS,UNKNOWN)
GYROMODE= '3' / number of gyros scheduled, T=3+OBAD
REFFRAME= 'GSC1' / guide star catalog version
MTFLAG = '' / moving target flag; T if it is a moving target

DATE-OBS= '2003-01-27' / UT date of start of observation (yyyy-mm-dd)
TIME-OBS= '15:20:01' / UT time of start of observation (hh:mm:ss)
EXPSTART= 5.266663890058E+04 / exposure start time (Modified Julian Date)
EXPEND = 5.266665048715E+04 / exposure end time (Modified Julian Date)
EXPTIME = 1000.00000 / exposure duration (seconds)--calculated
```

**APERTURE** – Portion of detector used during observation

**DATE-OBS** – The date of the start of the observation

**DEC\_TARG** – The declination of target in the night sky

**DETECTOR** – The detector used (WFC, HRC, or SBC)

**EXPSTART** – The exposure start time

**EXPFLAG** – The quality of the exposure

**EXPTIME** – The duration of the exposure

**FILTER** – The wavelength filter used during observation

**IMAGETYP** – The type of image (e.g. "EXTERNAL")

**OBSTYPE** – The type of observation (e.g. "IMAGING")

**proposal\_type** – The type of proposal (e.g. "CAL")

**PROPID** – The 5-digit proposal ID

**RA\_TARG** – The right-ascension of the target in the night sky

**rootname** – The 9-character unique identifier for the observation

**SUBARRAY** – Boolean value for if an image was a full-frame or not

**TARGNAME** – The name of the observation target (e.g. "Saturn")

**TIMEOBS** – The time of the start of the observation



## Data Structure: ACS FITS Filetypes and Rootnames

`<rootname>_<filetype>.fits`

Rootnames consist of a 9-character identifier corresponding to a unique observation

### Available filetypes:

**raw** - the raw, uncalibrated data that comes directly from HST

**f1t** - nominally calibrated data

**f1c** - nominally calibrated data corrected for Charge Transfer Efficiency (CTE) deficits.

**drz** - geometric distortion-corrected data

**drc** - geometric distortion-corrected and CTE corrected data

**spt** - telescope telemetry data

**j1t** - telescope pointing data

**j1f** - telescope drifting data

**crj** - cosmic ray rejected data

**crc** - cosmic ray rejected plus CTE corrected data

**asn** - observation association table.



## Intended Users and Use Cases

---

- The primary user of the `acsq1` application are ACS Instrument Analysts at STScI
- They use ACS data to perform instrument calibration and monitoring
- External users to STScI are possible since `acsq1` is non-proprietary

Use case 1: Visually inspect an image from the ACS archive

Use case 2: Determine which datasets exist in the ACS archive for a given set of observational parameters

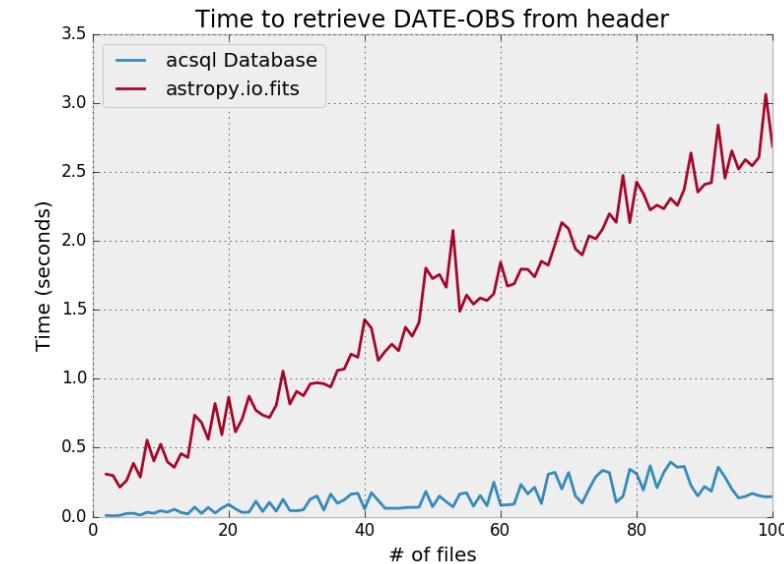
Use case 3: Determine the observational parameters for a given dataset

Use case 4: Programmatically analyze images across custom datasets



# Motivation

- Limitations in the FITS file structure (namely File I/O)
- Limitations in the current STScI data archive
  - Data retrieval latency
  - Data discovery
  - Data redundancy
- **acsq1** is a centrally-located instantly-accessible data archive with one-to-one correspondence with the ACS FITS file headers
  - No complicated data retrieval process
  - No need for file I/O
  - Intuitive database queries





# Implementation: Version Control, Programming Standards, and API Documentation

- Style Guide for Python (PEP8)
- Style Guide for Python Docstrings (PEP257)
- `numpydoc` Documentation Convention
- `git` + GitHub
- `readthedocs`

```
def get_proposal_type(proposid):
    """Return the ``proposal_type`` for the given ``proposid``.

    The ``proposal_type`` is the type of proposal (e.g. ``CAL``, ``GO``,
    etc.). The ``proposal_type`` is scraped from the MAST
    proposal status webpage for the given ``proposid``. If the
    ``proposal_type`` cannot be determined, a ``None`` value is returned.

    Parameters
    -----
    proposid : str
        The proposal ID (e.g. ``12345``).

    Returns
    -----
    proposal_type : int or None
        The proposal type (e.g. ``CAL``).
    """


```

`ingest.ingest.get_proposal_type(proposid)`

Return the `proposal_type` for the given `proposid`.

The `proposal_type` is the type of proposal (e.g. `CAL`, `GO`, etc.). The `proposal_type` is scraped from the MAST proposal status webpage for the given `proposid`. If the `proposal_type` cannot be determined, a `None` value is returned.

**Parameters:** `proposid` : str  
The proposal ID (e.g. `12345`).

**Returns:** `proposal_type` : int or None  
The proposal type (e.g. `CAL`).

The screenshot shows a GitHub repository page for `spacetelescope / acsql`. The repository has 228 commits, 2 branches, 7 releases, and 2 contributors. A recent merge pull request from `towson-paper` is visible. Below the repository stats, a list of commits shows changes to files like `acsq1`, `docs`, `paper`, `.gitignore`, `LICENSE`, `MANIFEST.in`, `README.md`, and `setup.py`. The bottom half of the image shows a generated documentation page for the `acsq1` package. It includes an `acsq1` section with a brief description, installation instructions, and dependency information (requiring `python 3.+`, `astropy`, and `flask`). The generated documentation also includes the `get_proposal_type` function documentation from the code block above.

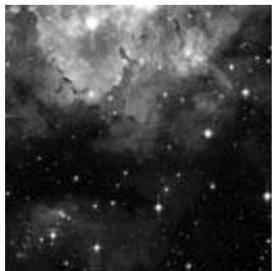


# Implementation: Filesystem

```
filesystem/
 jcp0/
    jcp001kwq/
        jcp001kwq_flc.fits
        jcp001kwq_flt.fits
        jcp001kwq_raw.fits
        jcp001kwq_spt.fits
        jcp001kwq_jif.fits
        jcp001kwq_jit.fits
    jcp001010/
        jcp001010_asn.fits
        jcp001010_drc.fits
        jcp001010_drz.fits
        jcp001010_jif.fits
        jcp001010_jit.fits
    jcp014tyq/
        ...
    jcp001kt1/
        ...
    ...
jcp3/
    ...
jcp7/
    ...
    ...
```

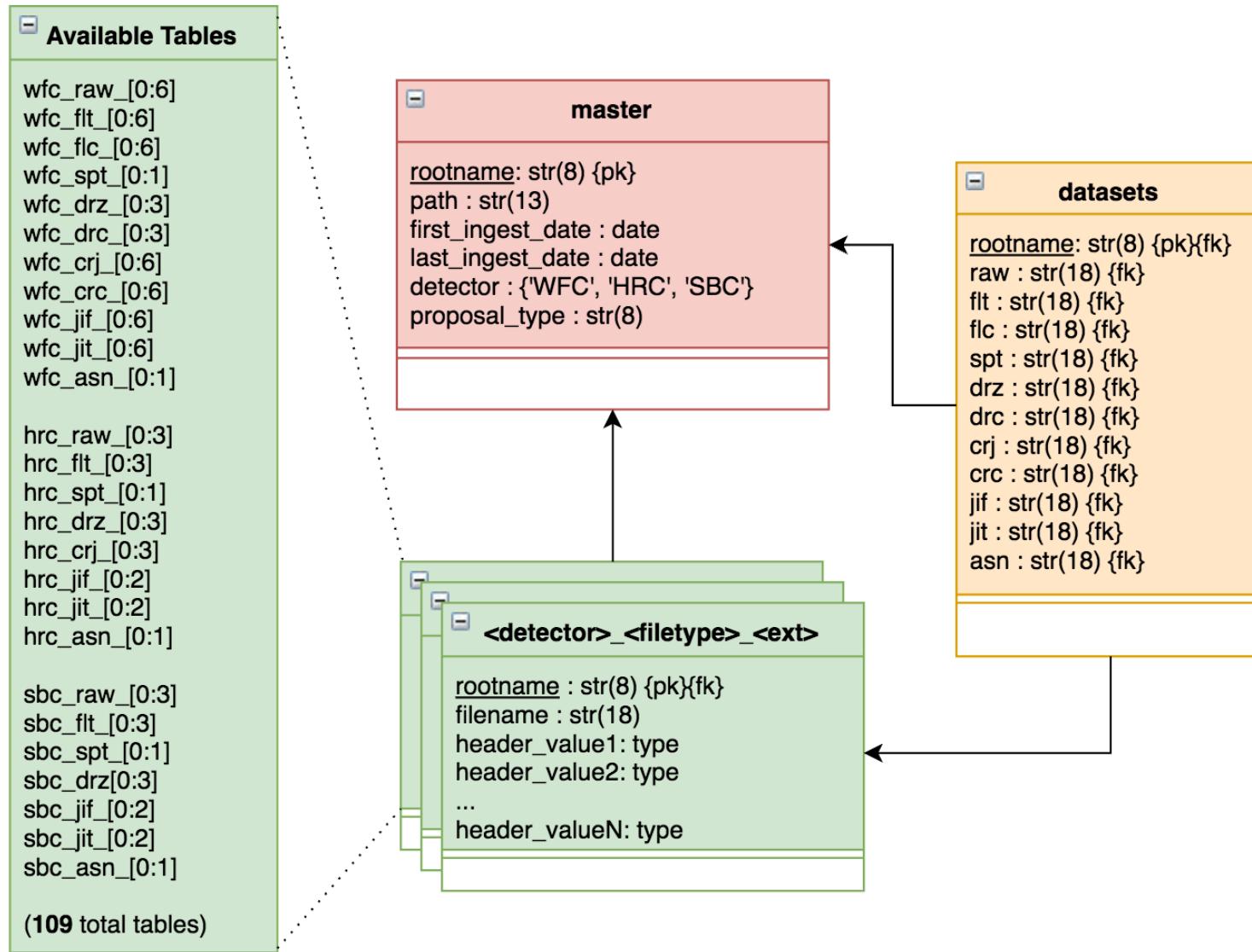
```
jpegs/
 12780/
    jbw901jtq_flc.jpg
    jbw901jtq_flt.jpg
    jbw901jtq_raw.jpg
    jbw901jxq_flc.jpg
    jbw901jxq_flt.jpg
    jbw901jxq_raw.jpg
    ...
 12781/
    jbx101f5q_flc.jpg
    jbx101f5q_flt.jpg
    jbx101f5q_raw.jpg
    jbx101f7q_flc.jpg
    jbx101f7q_flt.jpg
    jbx101f7q_raw.jpg
    ...
 12783/
    ...
 12787/
    ...
    ...

thumbnails/
 12780/
    jbw901jtq_flt.thumb
    jbw901jxq_flt.thumb
    ...
 12781/
    jbx101f5q_flt.thumb
    jbx101f7q_flt.thumb
    ...
 12783/
    ...
 12787/
    ...
    ...
```





# Implementation: Database Schema





# Implementation: MySQL and SQLAlchemy

- **SQLAlchemy**: Python SQL toolkit and Object Relational Mapper

```
class Master(base):
    """ORM for the master table."""
    def __init__(self, data_dict):
        self._dict_.update(data_dict)

    __tablename__ = 'master'
    rootname = Column(String(8), primary_key=True, index=True, nullable=False)
    path = Column(String(15), unique=True, nullable=False)
    first_ingest_date = Column(Date, nullable=False)
    last_ingest_date = Column(Date, nullable=False)
    detector = Column(Enum('WFC', 'HRC', 'SBC'), nullable=False)
    proposal_type = Column(Enum('CAL/ACS', 'CAL/OTA', 'CAL/STIS', 'CAL/WFC3',
                                'ENG/ACS', 'GO', 'GO/DD', 'GO/PAR', 'GTO/ACS',
                                'GTO/COS', 'NASA', 'SM3/ACS', 'SM3/ERO',
                                'SM4/ACS', 'SM4/COS', 'SM4/ERO', 'SNAP'),
                           nullable=True)
```

```
def load_connection(connection_string):
    """Return ``session``, ``base``, and ``engine`` objects for connecting to the ``acsq`` database.

    Create an ``engine`` using an given ``connection_string``. Create a ``base`` class and ``session`` class from the ``engine``. Create an instance of the ``session`` class. Return the ``session``, ``base``, and ``engine`` instances.

    Parameters
    -----
    connection_string : str
        The connection string to connect to the ``acsq`` database. The connection string should take the form:
        ``dialect+driver://username:password@host:port/database``

    Returns
    -----
    session : session object
        Provides a holding zone for all objects loaded or associated with the database.
    base : base object
        Provides a base class for declarative class definitions.
    engine : engine object
        Provides a source of database connectivity and behavior.
    """

    engine = create_engine(connection_string, echo=False, pool_timeout=100000)
    base = declarative_base(engine)
    Session = sessionmaker(bind=engine)
    session = Session()

    return session, base, engine
```



# Implementation: MySQL and SQLAlchemy

```
def orm_factory(class_name):
    """Create a SQLAlchemy ORM Classes with the given ``class_name``.

    Parameters
    -----
    class_name : str
        The name of the class to be created

    Returns
    -----
    class : obj
        The SQLAlchemy ORM
    """

    data_dict = {}
    data_dict['__tablename__'] = class_name.lower()
    data_dict['rootname'] = Column(String(8), ForeignKey('master.rootname'),
                                   primary_key=True, index=True,
                                   nullable=False)
    data_dict['filename'] = Column(String(18), nullable=False, unique=True)
    data_dict = define_columns(data_dict, class_name)
    data_dict['__table_args__'] = {'mysql_row_format': 'DYNAMIC'}

    return type(class_name.upper(), (base,), data_dict)
```

```
WFC_raw_0 = orm_factory('WFC_raw_0')
WFC_raw_1 = orm_factory('WFC_raw_1')
WFC_raw_2 = orm_factory('WFC_raw_2')
WFC_raw_3 = orm_factory('WFC_raw_3')
WFC_raw_4 = orm_factory('WFC_raw_4')
WFC_raw_5 = orm_factory('WFC_raw_5')
WFC_raw_6 = orm_factory('WFC_raw_6')

WFC_flt_0 = orm_factory('WFC_flt_0')
WFC_flt_1 = orm_factory('WFC_flt_1')
WFC_flt_2 = orm_factory('WFC_flt_2')
WFC_flt_3 = orm_factory('WFC_flt_3')
WFC_flt_4 = orm_factory('WFC_flt_4')
WFC_flt_5 = orm_factory('WFC_flt_5')
WFC_flt_6 = orm_factory('WFC_flt_6')

WFC_flc_0 = orm_factory('WFC_flc_0')
WFC_flc_1 = orm_factory('WFC_flc_1')
WFC_flc_2 = orm_factory('WFC_flc_2')
WFC_flc_3 = orm_factory('WFC_flc_3')
WFC_flc_4 = orm_factory('WFC_flc_4')
WFC_flc_5 = orm_factory('WFC_flc_5')
WFC_flc_6 = orm_factory('WFC_flc_6')
```



## Implementation: Data Ingestion Software Algorithm

---

1. Identify newly available data
2. Loop over **rootnames** in parallel
  3. Update **master** table with information about **rootname**
  4. Loop over available filetypes for given **rootname**
    5. Create Python dictionary with metadata about file
    6. For each FITS extension, extract header information and update **header** table
    7. Update **datasets** table
    8. If filetype is **raw**, **flt**, or **f1c**, then create JPEG
    9. If filetype is **flt**, then create Thumbnail

Encapsulated within `ingest.py`, `make_file_dict.py`, `make_jpeg.py`, `make_thumbnail.py`



# Implementation: Logging

```
08/15/2017 11:05:26 INFO: User: bourque
08/15/2017 11:05:26 INFO: System: plhstins1.stsci.edu
08/15/2017 11:05:26 INFO: Python Version: 3.5.2
08/15/2017 11:05:26 INFO: Python Path: /envs/anaconda3/envs/astroconda3/bin/python
08/15/2017 11:05:26 INFO: Numpy Version: 1.11.2
08/15/2017 11:05:26 INFO: Numpy Path: /site-packages/numpy
08/15/2017 11:05:26 INFO: Astropy Version: 1.2.1
08/15/2017 11:05:26 INFO: Astropy Path: /site-packages/astropy
08/15/2017 11:05:26 INFO: SQLAlchemy Version: 1.1.4
08/15/2017 11:05:26 INFO: SQLAlchemy Path: /site-packages/SQLAlchemy-1.1.4-py3.5-linux-x86_64.egg/sqlalchemy
08/15/2017 11:05:26 INFO: Gathering files to ingest
08/15/2017 11:06:00 INFO: j8zh21xv: Updated master table.
08/15/2017 11:06:01 INFO: j8zh21xv: Updated HRC_flt_0 table.
08/15/2017 11:06:01 INFO: j8zh21xv: Updated HRC_flt_1 table.
08/15/2017 11:06:01 INFO: j8zh21xv: Updated HRC_flt_2 table.
08/15/2017 11:06:01 INFO: j8zh21xv: Updated HRC_flt_3 table.
08/15/2017 11:06:01 INFO: j8zh21xv: Updated datasets table for flt.
08/15/2017 11:06:01 INFO: j8zh21xv: Creating JPEG
08/15/2017 11:06:02 INFO: j8zh21xv: Creating Thumbnail
08/15/2017 11:06:02 INFO: j8zh21xv: Updated HRC_raw_0 table.
08/15/2017 11:06:02 INFO: j8zh21xv: Updated HRC_raw_1 table.
08/15/2017 11:06:03 INFO: j8zh21xv: Updated HRC_raw_2 table.
08/15/2017 11:06:03 INFO: j8zh21xv: Updated HRC_raw_3 table.
08/15/2017 11:06:03 INFO: j8zh21xv: Updated datasets table for raw.
08/15/2017 11:06:03 INFO: j8zh21xv: Creating JPEG
08/15/2017 11:06:04 INFO: j8zh21xv: Updated HRC_spt_0 table.
08/15/2017 11:06:05 INFO: j8zh21xv: Updated HRC_spt_1 table.
08/15/2017 11:06:05 INFO: j8zh21xv: Updated datasets table for spt.
08/15/2017 11:06:05 INFO: j8zh21xv: End ingestion
```



## Implementation: Web Application

---

Built using Flask, a Python-based web framework

`acsq1_webapp.py` – Runs web application on `localhost` or server

`data_containers.py` – Various functions for gathering and returning data used by application

`form_options.py` – Stores data needed for database query form

`query_form.py` – Builds and validates database queries through web application

`query_lib.py` – Contains tools for executing database queries through web application

`static/css/*.css` – CSS templates

`static/js/*.js` – JavaScript functions

`templates/*.html` – HTML templates



# Implementation: acsql Package

```
acsql/
    __init__.py
    database/
        __init__.py
        database_interface.py
        make_tabledefs.py
        queries.py
        reset_database.py
        table_definitions/
            *.txt
        update_tabledefs.py
    ingest/
        __init__.py
        ingest.py
        make_file_dict.py
        make_jpeg.py
        make_thumbnail.py
    scripts/
        __init__.py
        ingest_production.py
    utils/
        __init__.py
        config.yaml
        utils.py
    website/
        __init__.py
        acsql_webapp.py
        data_containers.py
        form_options.py
        query_form.py
        query_lib.py
        static/
            css/
                *.css
            img/
                jpegs
                thumbnails
            js/
                *.js
        templates/
            *.html
```

The screenshot shows the GitHub repository page for `spacetelescope/acsql`. The page includes a summary bar with 228 commits, 2 branches, 7 releases, and 2 contributors. Below this is a list of recent commits:

Author	Commit Message	Date
bourque	Merge pull request #29 from spacetelescope/towson-paper ...	13 days ago
acsq1	Avoiding deprecation warning by switching to ints in indexing. Also s...	2 months ago
docs	Added discussion chapter.	a month ago
paper	Cleaning up some unnecessary files.	13 days ago
.gitignore	Added figure of acsql filesystem size over time.	a month ago
LICENSE	Updated LICENSE	8 months ago
MANIFEST.in	Updated manifest with new table definition text files.	5 months ago
README.md	Added requests dependency	2 months ago
setup.py	Adding other authors to the author list.	3 months ago

Below the commit history is a section titled **acsql** which contains the project's description and documentation link. It also lists dependencies, which include Python 3.+, astropy, and flask.

Available at <http://github.com/spacetelescope/acsql>



# Web Application: Homepage

The screenshot shows a web browser window titled "ACS Quicklook" with the URL "localhost:5000". The page has a dark header bar with the title "ACS Quicklook" and navigation links for "Database", "Archive", "GitHub", and "Documentation". The main content area is divided into two sections: "Database" on the left and "Archive" on the right. The "Database" section contains the text "Query the acsql Database" and a button labeled "Query form ». The "Archive" section contains the text "Links to public ACS archival data." and a button labeled "Archive »". At the bottom of the page, there is a link to "Space Telescope Science Institute".

ACS Quicklook

localhost:5000

Matthew

ACS Quicklook

Database Archive GitHub Documentation

## Database

Query the acsql Database

Query form »

## Archive

Links to public ACS archival data.

Archive »

Space Telescope Science Institute



# Web Application: Database Query Page

ACS Quicklook

localhost:5000/database/ Matthew

ACS Database Query Form

This form enables users to query the acsql database of all ACS images. Only a subset of the full database is represented here; if the options available do not meet your needs, you may enter a hard-coded MySQL command below.

**Rootname**

**Target Name**

**Proposal ID**  **Date Observed**  **Exposure Time**

**Proposal Type**  GO  GTO/ACS  CAL/ACS  SM3/ACS  SM3/ERO  SNAP  GO/PAR  GO/DD  GTO/COS  CAL/OTA  ENG/ACS  NASA  SM4/ACS  SM4/ERO  SM4/COS  CAL/WFC3  CAL/STIS

**Detector**  WFC  HRC  SBC

**Observation Type**  IMAGING  SPECTROSCOPC  CORONOGRAPHIC  INTERNAL

**Aperture**  WFC  WFC-FIX  WFC1  WFC1-1K

**Filter1**  F115LP  F122M  F125LP  F140LP

**Filter2**  F220W  F250W  F330W  F344N

**Image Type**  BIAS  DARK  FLAT  EXT

**PI Last Name**

**PI First Name**

**Output Columns**

<input type="checkbox"/> Rootname	<input type="checkbox"/> PI First Name	<input type="checkbox"/> Aperture	<input type="checkbox"/> Target Name	<input type="checkbox"/> Observation Mode
<input type="checkbox"/> Detector	<input type="checkbox"/> Proposal ID	<input type="checkbox"/> Expstart	<input type="checkbox"/> Target RA	<input type="checkbox"/> Subarray
<input type="checkbox"/> Proposal Type	<input type="checkbox"/> Filter1	<input type="checkbox"/> Date of Observation	<input type="checkbox"/> Target Dec	<input type="checkbox"/> Image Type
<input type="checkbox"/> PI Last Name	<input type="checkbox"/> Filter2	<input type="checkbox"/> Time of Observation	<input type="checkbox"/> Observation Type	<input type="checkbox"/> Association ID

**Output Format**

HTML table  CSV  Thumbnails



# Web Application: Query Results

The query returned 208 results

Show only visit: Show only target: Show only detector: Show only filter:

All visits All targets All detectors All filters

Sort by: default expstart exptime Filter logic: \*AND\* \*OR\* Clear

query\_results.csv

The query returned 208 results.

rootname	pr_inv_l	pr_inv_f	proposid	aperture	date_obs	time_obs	targname
jbxl50ck	Levay	Zolt	12812	WFC-FIX	2012-11-03	17:42:03	ANY
jbxl50co	Levay	Zolt	12812	WFC-FIX	2012-11-03	18:03:18	ANY
jbxl51en	Levay	Zolt	12812	WFC-FIX	2012-10-22	16:51:05	ANY
jbxl51er	Levay	Zolt	12812	WFC-FIX	2012-10-22	17:12:20	ANY
jbxl52z	Levay	Zolt	12812	WFC-FIX	2012-10-23	16:47:15	ANY
jbxl52k3	Levay	Zolt	12812	WFC-FIX	2012-10-23	17:08:30	ANY
jbxl53kw	Levay	Zolt	12812	WFC-FIX	2012-10-23	23:36:15	ANY
jbxl53l7	Levay	Zolt	12812	WFC-FIX	2012-10-24	00:51:28	ANY
jbxl54bf	Levay	Zolt	12812	WFC-FIX	2012-10-26	13:24:33	ANY
jbxl54bj	Levay	Zolt	12812	WFC-FIX	2012-10-26	13:45:48	ANY
jbxl55ez	Levay	Zolt	12812	WFC-FIX	2012-10-27	03:46:22	ANY
jbxl55f3	Levay	Zolt	12812	WFC-FIX	2012-10-27	04:07:37	ANY
jbxl56ht	Levay	Zolt	12812	WFC-FIX	2012-10-27	23:21:48	ANY
jbxl56i0	Levay	Zolt	12812	WFC-FIX	2012-10-28	00:30:13	ANY
jbxl57ac	Levay	Zolt	12812	WFC-FIX	2012-11-05	01:36:54	ANY
jbxl57ag	Levay	Zolt	12812	WFC-FIX	2012-11-05	01:58:09	ANY
jbxl58sp	Levay	Zolt	12812	WFC-FIX	2012-11-07	19:01:53	ANY
jbxl58st	Levay	Zolt	12812	WFC-FIX	2012-11-07	19:23:08	ANY
jchx01dj	Levay	Zolt	13623	WFC-FIX	2014-02-07	06:03:09	ANY
jchx01dl	Levay	Zolt	13623	WFC-FIX	2014-02-07	06:26:17	ANY
jchx01dr	Levay	Zolt	13623	WFC-FIX	2014-02-07	07:31:07	ANY
jchx01dv	Levay	Zolt	13623	WFC-FIX	2014-02-07	07:54:15	ANY
jchx02e6	Levay	Zolt	13623	WFC-FIX	2014-02-07	09:14:29	ANY
jchx02e8	Levay	Zolt	13623	WFC-FIX	2014-02-07	09:37:37	ANY
jchx02ee	Levay	Zolt	13623	WFC-FIX	2014-02-07	10:42:29	ANY
jchx02ei	Levay	Zolt	13623	WFC-FIX	2014-02-07	11:05:37	ANY
jchx03ib	Levay	Zolt	13623	WFC-FIX	2014-02-08	07:33:48	ANY
jchx03id	Levay	Zolt	13623	WFC-FIX	2014-02-08	07:56:56	ANY
jchx03ii	Levay	Zolt	13623	WFC-FIX	2014-02-08	09:02:01	ANY
jchx03im	Levay	Zolt	13623	WFC-FIX	2014-02-08	09:25:09	ANY
jchx04jz	Levay	Zolt	13623	WFC-FIX	2014-02-11	12:04:41	ANY
jchx04k1	Levay	Zolt	13623	WFC-FIX	2014-02-11	12:27:49	ANY
jchx04k6	Levay	Zolt	13623	WFC-FIX	2014-02-11	13:33:58	ANY

Thumbnails

HTML Table

query\_results.csv

```
1 |rootname,pr_inv_l,pr_inv_f,proposid,aperture,date_obs,time_obs,targname
2 jbxl50ck,Levay,Zolt,12812,WFC-FIX,2012-11-03,17:42:03,ANY
3 jbxl50co,Levay,Zolt,12812,WFC-FIX,2012-11-03,18:03:18,ANY
4 jbxl51er,Levay,Zolt,12812,WFC-FIX,2012-10-22,16:51:05,ANY
5 jbxl51en,Levay,Zolt,12812,WFC-FIX,2012-10-22,17:12:20,ANY
6 jbxl52jz,Levay,Zolt,12812,WFC-FIX,2012-10-23,16:47:15,ANY
7 jbxl52k3,Levay,Zolt,12812,WFC-FIX,2012-10-23,17:08:30,ANY
8 jbxl53kw,Levay,Zolt,12812,WFC-FIX,2012-10-23,23:36:15,ANY
9 jbxl53l7,Levay,Zolt,12812,WFC-FIX,2012-10-24,00:51:28,ANY
10 jbxl54bf,Levay,Zolt,12812,WFC-FIX,2012-10-26,13:24:33,ANY
11 jbxl54bj,Levay,Zolt,12812,WFC-FIX,2012-10-26,13:45:48,ANY
12 jbxl55ez,Levay,Zolt,12812,WFC-FIX,2012-10-27,03:46:22,ANY
13 jbxl55f3,Levay,Zolt,12812,WFC-FIX,2012-10-27,04:07:37,ANY
14 jbxl56ht,Levay,Zolt,12812,WFC-FIX,2012-10-27,23:21:48,ANY
15 jbxl56i0,Levay,Zolt,12812,WFC-FIX,2012-10-28,00:30:13,ANY
16 jbxl57ac,Levay,Zolt,12812,WFC-FIX,2012-11-05,01:36:54,ANY
17 jbxl57ag,Levay,Zolt,12812,WFC-FIX,2012-11-05,01:58:09,ANY
18 jbxl58sp,Levay,Zolt,12812,WFC-FIX,2012-11-07,19:01:53,ANY
19 jbxl58st,Levay,Zolt,12812,WFC-FIX,2012-11-07,19:23:08,ANY
20 jchx01dj,Levay,Zolt,13623,WFC-FIX,2014-02-07,06:03:09,ANY
21 jchx01dl,Levay,Zolt,13623,WFC-FIX,2014-02-07,06:26:17,ANY
22 jchx01dr,Levay,Zolt,13623,WFC-FIX,2014-02-07,07:31:07,ANY
23 jchx01dv,Levay,Zolt,13623,WFC-FIX,2014-02-07,07:54:15,ANY
24 jchx02e6,Levay,Zolt,13623,WFC-FIX,2014-02-07,09:14:29,ANY
25 jchx02e8,Levay,Zolt,13623,WFC-FIX,2014-02-07,09:37:37,ANY
26 jchx02ee,Levay,Zolt,13623,WFC-FIX,2014-02-07,10:42:29,ANY
27 jchx02ei,Levay,Zolt,13623,WFC-FIX,2014-02-07,11:05:37,ANY
28 jchx03ib,Levay,Zolt,13623,WFC-FIX,2014-02-08,07:33:48,ANY
29 jchx03id,Levay,Zolt,13623,WFC-FIX,2014-02-08,07:56:56,ANY
30 jchx03ii,Levay,Zolt,13623,WFC-FIX,2014-02-08,09:02:01,ANY
31 jchx03im,Levay,Zolt,13623,WFC-FIX,2014-02-08,09:25:09,ANY
32 jchx04jz,Levay,Zolt,13623,WFC-FIX,2014-02-11,12:04:41,ANY
33 jchx04k1,Levay,Zolt,13623,WFC-FIX,2014-02-11,12:27:49,ANY
34 jchx04k6,Levay,Zolt,13623,WFC-FIX,2014-02-11,13:33:58,ANY
```

Line 1, Column 1

Spaces: 4 Plain Text

CSV File



# Web Application: ACS Archive

The screenshot shows a web browser window titled "ACS Quicklook" with the URL "localhost:5000/archive/". The page has a dark header bar with the title and a navigation menu on the right containing "Database", "Archive" (which is highlighted in red), "GitHub", and "Documentation". The main content area is titled "ACS Archive" and displays a table of data. The table has 25 rows, each representing a record with 13 columns. The columns are: RowID, Column1, Column2, Column3, Column4, Column5, Column6, Column7, Column8, Column9, Column10, Column11, Column12, and Column13. The data values are mostly numerical, ranging from 8183 to 9027.

8183	9503	9830	10188	10459	10626	10907	11655	12250	12759	13407	13942
8947	9558	9831	10189	10460	10627	10909	11658	12253	12760	13410	13945
8948	9560	9835	10190	10461	10628	10910	11663	12254	12780	13412	13952
8992	9562	9836	10192	10466	10629	10911	11669	12256	12781	13419	13953
8993	9563	9837	10195	10471	10630	10913	11670	12257	12782	13420	13954
9005	9564	9838	10196	10472	10631	10915	11675	12262	12783	13422	13955
9006	9565	9842	10198	10473	10632	10917	11676	12270	12787	13425	13956
9008	9566	9847	10199	10474	10633	10918	11677	12271	12788	13433	13957
9009	9567	9851	10200	10475	10634	10919	11679	12273	12789	13435	13958
9010	9568	9853	10201	10476	10635	10920	11681	12282	12790	13437	13959
9011	9574	9854	10204	10483	10695	10921	11683	12286	12791	13439	13960
9012	9575	9856	10205	10486	10696	10922	11684	12292	12811	13441	13961
9013	9578	9857	10206	10487	10697	10923	11688	12297	12812	13442	13962
9014	9583	9860	10207	10488	10698	10991	11689	12309	12817	13443	13964
9015	9584	9861	10210	10489	10701	10992	11691	12310	12858	13449	13965
9016	9586	9862	10213	10490	10703	10994	11695	12311	12860	13459	13966
9017	9587	9863	10214	10491	10705	10996	11704	12313	12866	13461	14034
9018	9588	9864	10216	10492	10707	10997	11710	12317	12871	13463	14035
9019	9647	9869	10217	10493	10710	11003	11711	12323	12872	13470	14037
9020	9648	9872	10227	10494	10711	11004	11713	12326	12875	13474	14038
9022	9649	9873	10228	10495	10713	11005	11715	12327	12877	13476	14039
9023	9650	9877	10229	10496	10715	11008	11718	12328	12878	13479	14056
9024	9651	9884	10231	10497	10718	11011	11722	12331	12880	13480	14057
9025	9652	9889	10233	10498	10720	11013	11724	12362	12884	13483	14061
9026	9654	9890	10235	10499	10722	11015	11731	12367	12908	13485	14063
9027	9655	9891	10237	10500	10729	11020	11734	12368	12911	13493	14072
9028	9656	9892	10239	10502	10730	11021	11735	12369	12912	13495	14074



# Web Application: View Proposal

ACS Quicklook    ACS Quicklook

localhost:5000/archive/14039/ Matthew

ACS Quicklook Database Archive GitHub Documentation

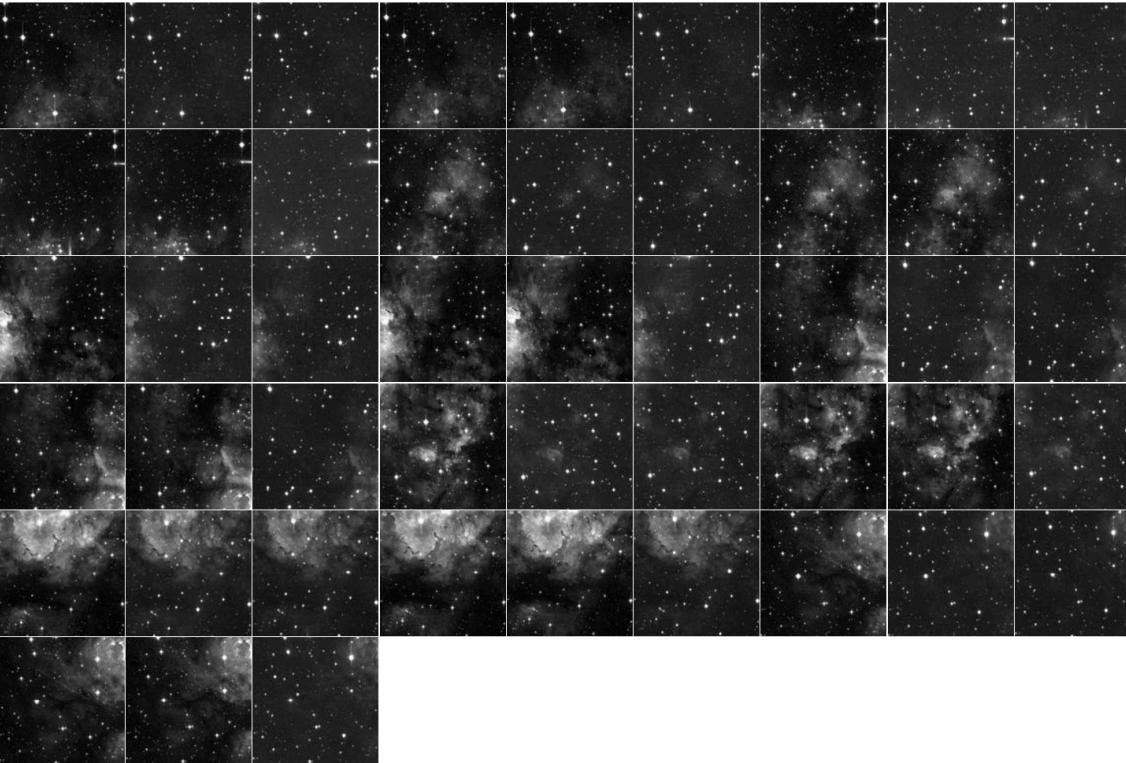
**Proposal 14039** Broad-band imaging of Westerlund 2

Cycle 22, completed  
48 images, 8 visits  
[Proposal Info](#)

Show only visit: All visits   Show only target: All targets   Show only detector: All detectors   Show only filter: All filters

Sort by:  default  expstart  exptime

Filter logic:  "AND"  "OR" [Clear](#)



Space Telescope Science Institute



# Web Application: View Image

ACS Quicklook   ACS Quicklook   ACS Quicklook   Matthew

localhost:5000/archive/14039/jcs718kmq/

ACS Quicklook Database Archive GitHub Documentation

### jcs718kmq

Proposal 14039: Broad-band imaging of Westerlund 2  
Image 40 of 48  
No FITS downloads available yet  
View all in [proposal](#); view [headers](#); view in [JS9](#)  
View JPEG for [f1c raw](#)

Expstart: 56991.84858445  
Filter: CLEAR1L/F814W  
Aperture: WFCENTER  
Exposure time: 348.0

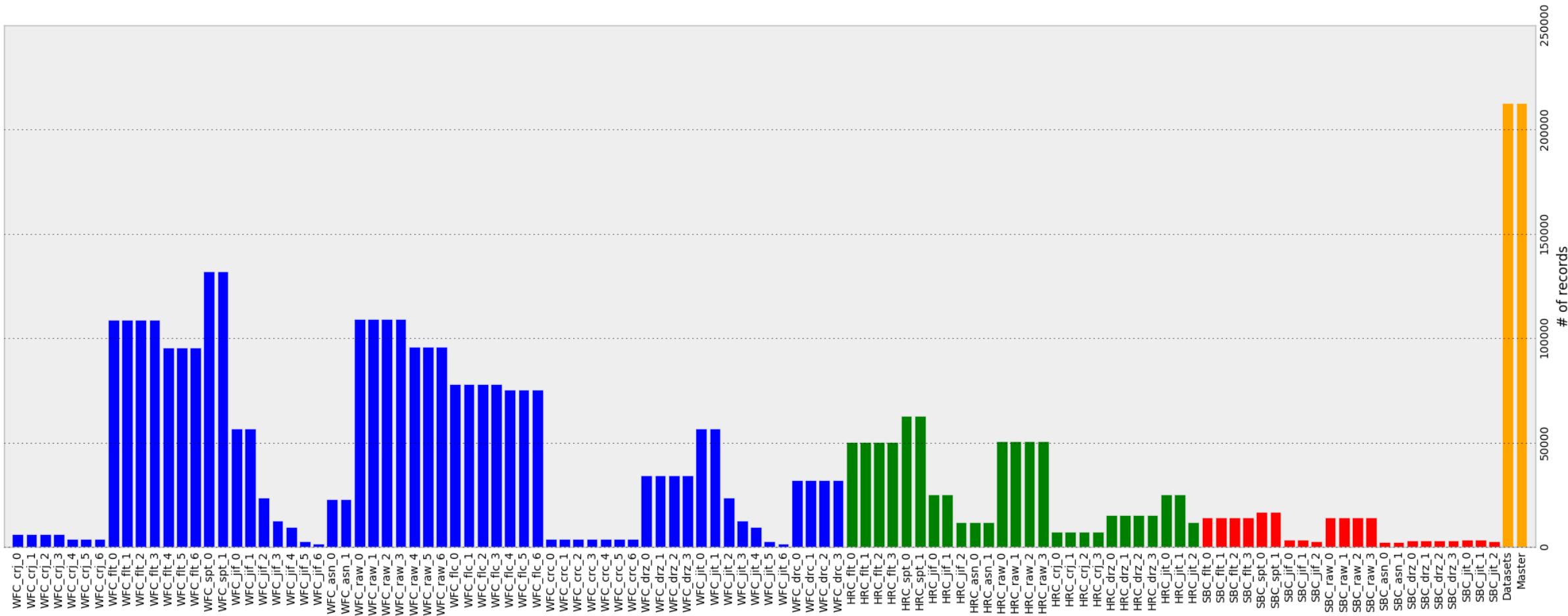
Expflag: NORMAL, Quality: None  
Coordinates: 156.02387, -57.76315  
Target: WESTERLUND-2  
PI: Zolt Levay



Space Telescope Science Institute



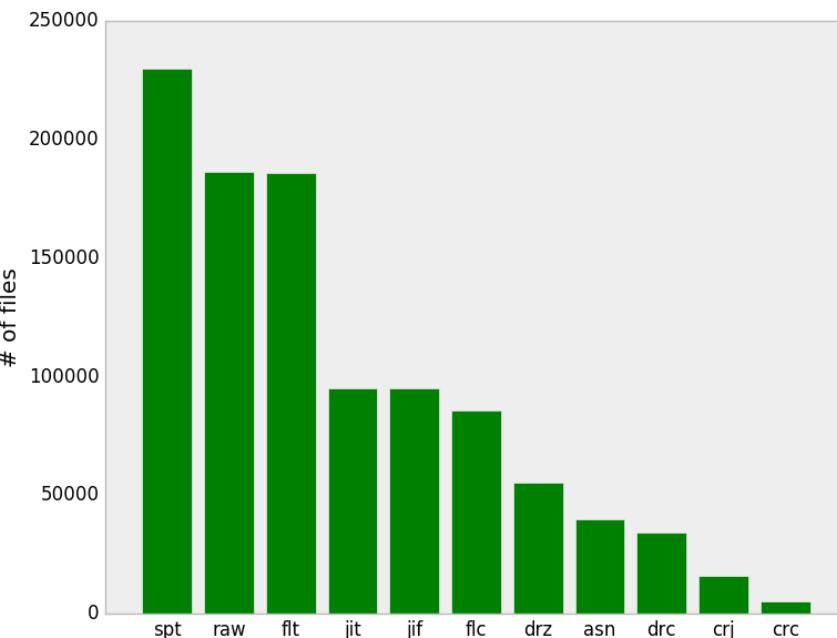
# Deliverables



## Database Deliverable



# Deliverables



Filesystem Deliverable

A screenshot of a GitHub repository page for "spacetelescope / acsql". The repository has 228 commits, 2 branches, 7 releases, and 2 contributors. The latest commit was 13 days ago. The repository description is "The Advanced Camera for Surveys Quicklook Project".

Software Package Deliverable

A screenshot of the acsql documentation website. It features a sidebar with links for Database, Ingest, Utils, and Website. The main content area displays the "The Hubble Space Telescope (HST) Advanced Camera for Surveys (ACS) Quicklook Project (acsql)" documentation, which includes a brief description and a table of contents for the codebase.

API Documentation Deliverable

A screenshot of the ACS Quicklook web application. It shows a grayscale astronomical image of a star-forming region. The interface includes various metadata fields such as "Proposal: 1402B", "Filter: CLEARWFLR14W", "Aperture: WFC3/IR", "Exptime: 348.0", and "Target: WESTERlund-2". Below the image, it says "Space Telescope Science Institute".

Web Application Deliverable



## Acknowledgements

---

- The ACS Instrument Team: Roberto Avila, Andreas Bellini, Ralph Bohlin, David Borncamp, Marco Chi(something) Dan Coe, Tyler Dehjardins, Norman Grogin, Samantha Hoffman, Pey-Lian Lim, Ray Lucas, Aparna Maybhate, Matt McMaster, Vera Platais, Jenna Ryon
- The WFC3/Quicklook Team: Varun Bajaj, Ariel Bowers, Michael Dulude, Meredith Durbin, Jules Fowler, Catherine Gosmeyer, Heather Gunning, Harish Khandrika, Catherine Martlin, Jennifer Medina, Abhi Rajan, Clare Shanahan, Ben Sunquist, Alex Viana
- Other Employees of STScI: Elena Sabbi, Francesca Boffi, Alex Yermolaev, Fausat Ogunsanya, Vera Gibbs
- Paper Reviewers: Sara Ogaz and Jules Fowler