Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

The HST BestRef System - Determining the proper calibration reference files for an HST exposure

Michael S. Swam, Steve Lubow, Lee Hurt

Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218

Abstract. This paper will describe the software architecture of the Hubble Space Telescope (HST) BestRef system, which is used to determine the proper calibration reference files for a particular HST exposure. The system, which makes use of the HST Calibration Database System (CDBS), is highly table-driven, allowing additions of new reference file types to support continual instrument calibration development as well as each new science instrument added during HST servicing missions. The system also works in concert with the HST Archive Catalog database and its StarView and web-based user interfaces to supply archive users with the set of proper reference files for any exposure they might retrieve from the archive.

1. The Purpose of HST BestRef

The HST BestRef system serves two masters: the science data processing pipelines and the archive catalog. The pre-archive pipeline receives science telemetry from HST and proceses the data into FITS files for the archive system (Swade, Hopkins, & Swam 2001). The On-The-Fly Reprocessing (OTFR) pipeline regenerates these science data products at archive retrieval time, by starting again from the base HST science telemetry (Swam, Hopkins, & Swade 2001). Both pipelines rely on BestRef to compute the best reference files for a given HST exposure so that these files can be applied during calibration. The archive catalog relies on BestRef to perform updates to the stored list of best reference files held for each archived HST exposure. When a science instrument team produces new reference files they often apply to some subset of the existing archive content. This subset of exposures must be identified so that the new reference file can replace the out-dated recommendation in the catalog. Using the BestRef system to serve both of these masters ensures that both receive identical recommendations given the same inputs.

2. The BestRef Architecture

The major components of the BestRef software architecture are the CDBS, the BestRef rules, and the BestRef rules engine.

2.1. CDBS

For every reference file created for a HST science instrument, a file description is installed in the CDBS (Cox, Lubow, & Tullos, 1998). This description includes the instrument mode parameters (e.g. detector id, filter id, etc.) that define the types of exposures the reference file applies to. This information is held in separate database tables for each HST science instrument, and each table contains fields for the union of all of the mode-selection parameters across that instrument's reference file set. For any particular table row that documents one mode of a particular reference file type, there are likely some mode fields that do not apply to that type and these are given obvious phony values in that row.

Also part of the reference file description is a "use-after" date that marks the earliest date-time after which this reference file should be used. When a delivery is made to the CDBS for a new reference file that exactly covers the same modes and use-after date as a previously delivered file, the previous file is marked "rejected". The rejected file is not deleted from the database, which maintains historical tracking, but is flagged as inactive, marked with the rejecting filename and date, and will never be recommended by BestRef again. Given these detailed descriptions of reference file content and their application modes, it is possible to write relatively simple database queries that locate the proper reference file of a given type for a given HST exposure.

2.2. The BestRef Rules

All of the BestRef reference file selection rules for all HST instruments are stored in a file of XML. These rules indicate the search criteria to use in selecting a particular reference file type for a particular HST science instrument. They include file selection and file restriction rules, both of which can include conditionals (e.g. "only use central wavelength to select a STIS flat-field when in SPECTROSCOPIC mode", or "only use the STIS image distortion correction file in IMAGING mode"). The rules also contain the mapping from FITS header keyword name to the name of a particular reference file type, and a flag indicating if a reference file is required or optional (only failure to fill a required file will generate an error condition). The rules are read by the BestRef rules engine once at task start-up, and are held in memory as each selected science exposure has its reference files computed. Here is a cutout example of the XML format for the STIS Image Distortion Correction table:

```
<!--image distortion correction table -->
<REFFILE>
    <REFFILE_TYPE> IDC </REFFILE_TYPE>
    <REFFILE_KEYWORD> IDCTAB </REFFILE_KEYWORD>
    <REFFILE_FORMAT> TABLE </REFFILE_FORMAT>
    <FILE_SELECTION>
       <FILE_SELECTION_FIELD> DETECTOR </FILE_SELECTION_FIELD>
    </FILE_SELECTION>
    <RESTRICTION>
       <RESTRICTION_TEST> (aSource._keywords['OBSTYPE'][0] == 'IMAGING')
       </RESTRICTION_TEST>
    </RESTRICTION>
</REFFILE>
```

Note the format of the "RESTRICTION-TEST" rule. It is actually written in Python code, which is directly read and executed by the rules engine. This provides flexibility for adding rules consisting of any Python construct that can be evaluated to a logical (True/False) value. This same mechanism is used to add a "FILE-SELECTION-TEST" as shown below, which only uses a field for file selection (e.g. CCDAMP) if certain conditions are met (e.g. the DETECTOR is NOT "SBC"), otherwise the field does not contribute to the selection criteria.

```
<FILE_SELECTION>
  <FILE_SELECTION_FIELD> CCDAMP </FILE_SELECTION_FIELD>
  <FILE_SELECTION_TEST> (aSource._keywords['DETECTOR'][0] != 'SBC')
  </FILE_SELECTION_TEST>
</FILE_SELECTION>
```

By implementing the reference file selection criteria in this rules format, it is quite simple to add entries to support new reference file types or to alter the selection criteria for existing types, and it is also possible to dynamically build the CDBS queries in a table-driven manner.

2.3. The BestRef Rules Engine

The program that executes the BestRef rules is the rules engine. This task consists of 5100 lines of object-oriented Python code across 15 modules. The rules engine can obtain the rule inputs from either FITS header keywords (in pipeline mode) or database fields (in archive catalog mode). The PyFITS Python FITS software (Hsu and Hodge 2004) is used for the FITS interface, while the STPyDB module[1] provides a Python API to the Sybase databases at the Space Telescope Science Institute (STScI). Reference file names are computed by applying the rules to the rule inputs and generating database queries to the CDBS. The resulting reference file names can be stored in FITS keywords or written to a file of SQL that is later applied the archive catalog. The design choice of using rules and a rules engine allows new reference file types that require standard rule matching to be added without changing the rules engine. Should selection of a particular reference file type require a more complex algorithm, again no change is needed to the rules engine as the design allows a rule to call special case procedures, again written in Python, as shown below:

```
<REFFILE_FUNCTION> filename = cdbs_db.wfpc2_flatfile( reff, aSource)
</REFFILE_FUNCTION>
```

In this case, the WFPC-2 flatfield reference file requires both filter ids in the selection query, but the filter order is irrelevant. This is handled internally by the special case procedure. Currently only three (3) special procedures are used throughout all of the rules supporting the current HST instruments (ACS, STIS, NICMOS, WFPC-2) and the next generation instruments (COS, WFC3), so the vast majority of reference file types can be selected using standard rule matching.

---

[1] http://www.stsci.edu/spst/UnixTransition/doc/stpydb.html

3. BestRef in use

The flexibility of this architecture and implementation allows a single set of code to satisfy the various requirements for computing the set of reference files for a HST exposure. In pipeline mode, the BestRef task is run just before calibration, and fills the FITS header reference file keywords with the names of the reference files to use during calibration. In archive catalog mode, a BestRef job is invoked periodically to read the list of newly delivered reference files from the CDBS and assign these new files to existing archived HST exposures. When an archive user requests the set of best reference files be returned to them along with the reprocessed science data, database records in the archive catalog are read by the catalog user interface (StarView[2] or MAST[3]) to obtain the reference file recommendations filled by BestRef.

4. Future Plans

While the BestRef system is performing well in operations, some adjustments are planned to address issues of timing. As described, the archive catalog BestRef is invoked on a periodic basis to look for new reference file deliveries, but if such a delivery occurs long before the next periodic invocation then contradictory recommendations can occur. The OTFR system will always return science exposures calibrated using the most up-to-date set of reference files (which it obtains from the pipeline BestRef), but the archive catalog can return an older set of reference files to the user's disk since its tables have not yet had the benefit of the periodic BestRef update. Plans are to change the periodic update into a direct invocation of the archive catalog BestRef when a new reference file delivery occurs.

References

Cox, C. R., Lubow, S. & Tullos, C. 1998, in SPIE Proc., Vol. 3349, 218

Hsu, J. C. & Hodge, P. 2004, this volume, 828

Swade, D. A., Hopkins, E. & Swam, M. S. 2001, in ASP Conf. Ser., Vol. 238, ADASS X, ed. F. R. Harnden, Jr., F. A. Primini, & H. E. Payne (San Francisco: ASP), 295

Swam, M. S., Hopkins, E. & Swade, D. A. 2001, in ASP Conf. Ser., Vol. 238, ADASS X, ed. F. R. Harnden, Jr., F. A. Primini, & H. E. Payne (San Francisco: ASP), 291

[2] `http://starview.stsci.edu/`

[3] `http://archive.stsci.edu/`