Internal Memo

Title: DFO Data Association II. The Association Block

From: Reinhard Hanuschik
To: DFO team, DFS team
Date: 2003-07-21
Purpose: Description of Association Block

Change record:

version 1.0	2003-01-23
version 1.1	2003-02-12: split into AssocBlock description, plus dfo comments included
version 1.2	2003-07-21: AB structure updated

0 Applicable documents

[1] DFO Data Association. I. DFO Association Rules (R. Hanuschik): Internal Memo to DFO group (version 1.3, 2003-07-21);
[2] DFO Data Association. III. General tools (R. Hanuschik): Internal Memo to DFO group (version 1.2, 2003-07-21)

Specific nomenclature is explained in [1].

1 Association Blocks

This document is part of a set of documents dealing with DFO data association. In [1] the general rules and concepts have been described. In this document the Association Block as the central interface is described. The third document [2] deals with specifications for a set of tools which creates and updates Association Blocks.

A key concept in the framework of DFO data association is the Association Block (AB). An AB collects all association information about a single raw file, or a set of logically connected raw files. It is the extension of the Reduction Block (RB) concept. It is more general since it is equally applicable to instrument modes being supported or not supported by pipelines. It is also more general since it collects association information both useful for processing and for packing.

The Association Block is *created* and *updated* by a set of general association tools which are described in a separate document [2]. They are *read* and *interpreted* by instrument-specific tools (scripts) which e.g. provide the translation of an AB into an RB or into a command line used for launching a pipeline recipe. Hence they are the interface between common (instrument-independent) tools and specific tools.

Association Blocks can also be regarded as the interface between operator and machine for all association processes. They provide the translation of abstract association rules into a human-readable form and vice versa.

This document describes the proposed structure of Association Blocks.

2 Association maps

Association maps are the graphical representation of association rules for a specific instrument mode. They have structural elements which are reflected in Association Blocks. Therefore they are shortly presented here. Details of association maps are described in [1].

An example association map (GIRAFFE, MEDUSA MOS mode) is shown in Figure 1. In this example, there are three types of raw calibration frames, BIAS, FLAT and WAVE, and one type of SCIENCE frame.

- The BIAS frames are processed by a pipeline into master BIAS products. BIAS frames to be combined have to match in the BIN parameter (*raw match key* 1). Since master BIAS frames are used to create other products, BIAS frames need to be processed first.
- FLAT frames to be combined have to match in BIN, CALS_ID, PLATE_ID and WLEN (*raw match key* 2). They need a master BIAS frame which associates through *product match key* 1 (BIN).
- WAVE frames follow the same rules as FLAT frames. They are processed into dispersion solution tables. The processing order of WAVE and FLAT frames in principle does not matter, but they need to be processed after the BIAS frames.

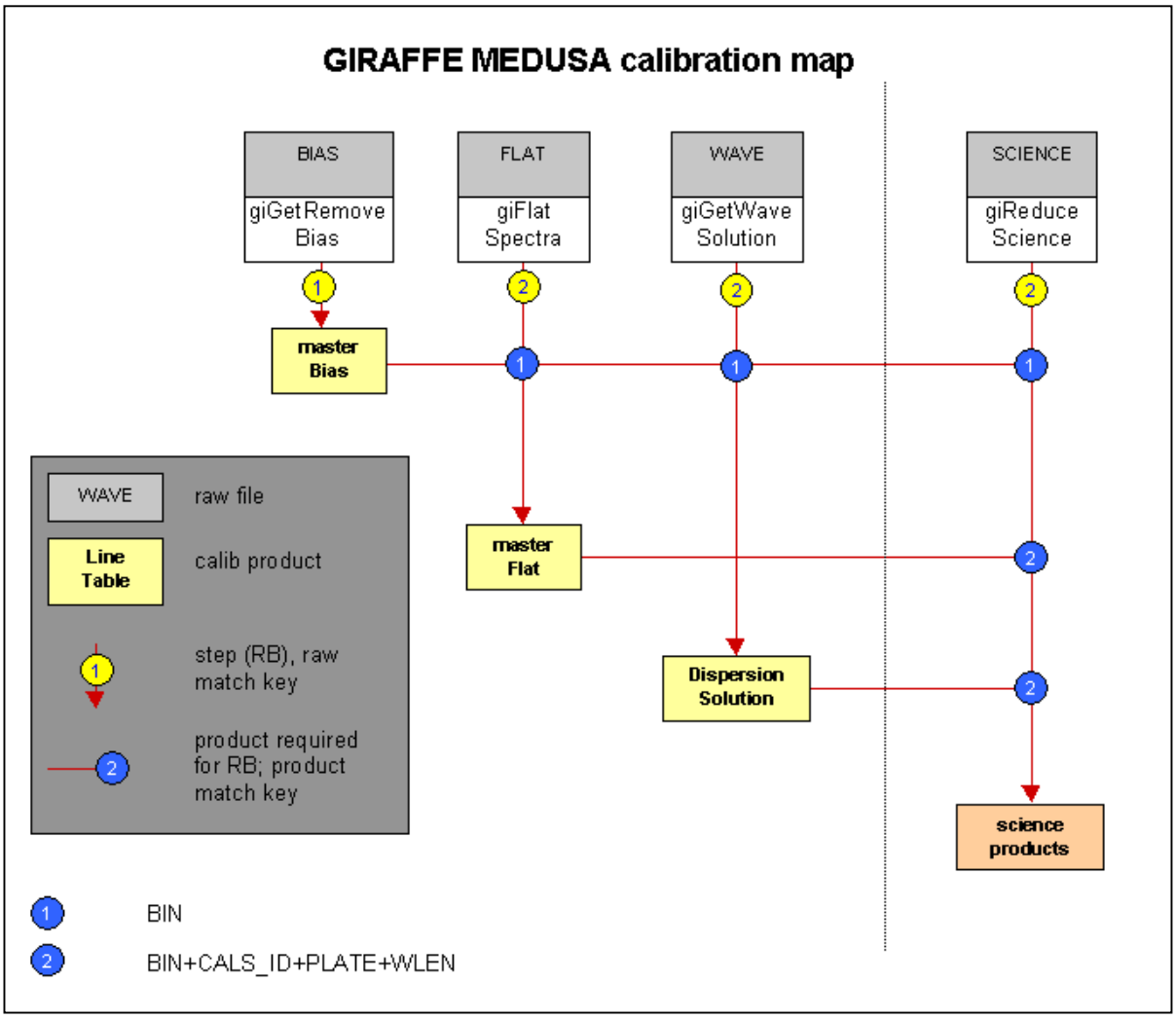

Figure 1. The association map for GIRAFFE (MEDUSA mode).

- The final step in the cascade is the processing of SCIENCE raw frames. These have to match by raw match key 2, they need to find a master BIAS matching product key 1, and a master FLAT and a dispersion table matching product key 2.

3 Structure of Association Blocks

In the association map in Figure 1, each column (raw file type) can create a certain kind of Association Block. Actually, there will be one AB per *primary raw file* [1]. For each raw type, there are *n* different basic data sets that are defined by the value of their raw match key. The simplest example for GIRAFFE are BIAS frames. They come in just one binning and read mode (1x1) so usually there will be one BIAS AB per night. FLATs and WAVEs may split into 150 basic data sets, different by WLEN, PLATE_ID etc.

3.1 General structure

Each AB carries the name of the parent primary file, with extension `ab`. It is an ASCII file. It has comment lines (starting with #) and entry lines. The entry lines all start with a *qualifier,* followed by one or several *value fields*. The fields contain file names, path names, or values.

An AB has the following general components:

- an initial section with versioning information
- some general information including AB name, parent event and completeness flag
- the raw match key used
- the root name of products (if the mode is pipeline-supported)
- Reduction Block name and reduction log name
- list of input raw files (the primary file, associated raw files)
- additional raw files (used for packing)
- list of product files
- names, types of associated calibration product files
- additional calibration product files (used for packing)
- Reduction Block parameters
- associated QC1 parameters
- associated graphical information
- a status section
- an RB section containing the equivalent Reduction Block (if the mode is pipeline-supported)

The AB components are presented in detail in the following. A complete AB is found in the Appendix.

The string NONE is always written when no value is known or applicable.

An important part of the AB concept is that it has all components created in the beginning (before data processing) but some of them are void. Only after the final processing stage in the daily DFO workflow (QC checks and final distribution) all information is available. This is why then an existing AB is updated and the missing information is filled in. Once the AB is completed, it can be stored, read for packing etc.

3.2 Path names and file names

Since the concept of ABs is such that they provide all information necessary for association, they need to provide path names in addition to file names, at least in those cases when the path name is not known by other conventions. Such convention could be that e.g. processing logs are always stored in $DFO_LOG_DIR which is then not written explicitly into the AB.

Whenever path names are written into the AB, they should follow DFO-wide global variables rather than explicit directory names. This convention is necessary since DFO installations are known to be variable on timescales of months or years. The set of DFO global variables needs to be defined yet. In the following it is assumed that $DFO_RAW_DIR, $DFO_CAL_DIR and $DFO_RED_DIR are known to the system and do exist.

In the following the AB sections are defined.

3.3 Versioning section

Description:	• tool version • configuration file version The tool version is read from findAssoc. Configuration file version is a parameter in config.assoc.
Filled:	on creation
Qualifier(s):	`TOOL_VERSION | CONFIG_VERSION`
Example:	`TOOL_VERSION 1.0` `CONFIG_VERSION giraffe_1.0.0`

3.4 General information

Description:	• the instrument name • the date (defining the raw data pool) • DPR_CATG, RAW_TYPE • the pipeline recipe name (if existing, otherwise NONE) General information about the AB: • the AB name • the AB event • a completeness flag The completeness flag describes whether all master calibrations necessary to process have been found. • OBS_PROG_ID, OB_ID, MJD-OBS, MJD-OBS_MODE (used for packing of science data) MJD-OBS is the time of the begin of observation; MJD-OBS_MODE is the time at the middle of the observation; this parameter is evaluated for the time matching rules
Filled:	on creation
Qualifier:	INSTRUMENT | DATE DPR_CATG |RAW_TYPE | RECIPE AB_NAME|AB_EVENT|COMPLETENESS OBS_PROG_ID |OBS_ID MJD-OBS | MJD-OBS_MODE
Example:	INSTRUMENT giraffe DATE 2003-01-24 DPR_CATG SCIENCE RAW_TYPE SCIENCE RECIPE giReduceScience AB_NAME GIRAF.2003-01-25T05:48:19.026.ab AB_EVENT SINGLE COMPLETENESS COMPLETE OBS_PROG_ID 60.A-9147(A) OBS_ID 132137 MJD-OBS 52664.24188688 MJD-OBS_MOD 52664.24192314

3.5 Match key information

Description:	The used match key value is written into the AB, mainly to ease monitoring.
Filled:	on creation
Qualifier:	RAW_MATCH_KEY
Example:	RAW_MATCH_KEY BIN=BIN1 RAW_MATCH_KEY INS_SLIT_NAME=Medusa1 RAW_MATCH_KEY INS_GRAT_ID=HR RAW_MATCH_KEY INS_GRAT_WLEN=WLN651.5

3.6 Product root name

Description:	This information is used to predict the names of all associated result

	files (*virtual calibration products*). Virtual file names are used upon filling the calibration cascade. This field is relevant for pipeline-supported modes only.
Filled:	on creation
Qualifier:	RBS_ROOT_NAME
Example:	RBS_ROOT_NAME r.GIRAF.2003-01-25T05:48:19.026

3.7 RB_NAME, LOG_NAME

Description:	If the AB has a successfully run RB, the RB name is listed along with the log name. Otherwise filled by NONE.
Filled:	on updating
Qualifier:	RB_NAME | LOG_NAME
Example:	RB_NAME GIRAF.2003-01-25T05:48:19.026.rX LOG_NAME GIRAF.2003-01-25T05:48:19.026.rblog

3.8 List of input raw files

Description:	Here all raw files are listed which are input for processing. This list has at least one file name, the name of the primary raw file, but may contain more than one. Apart from the qualifier, it has two components: path and file name(s), DO_CATG. The complete path to the file is given, using DFO global variables.
Filled:	on creation
Qualifier:	RAWFILE
Example:	RAWFILE $DFO_RAW_DIR/2003-01-24/GIRAF.2003-01-25T05:48:19.026.fits SCI_COMB_MOS

3.9 List of associated raw files

Description:	This list has those additional raw files (if any) which are associated for packing, but not for processing. A typical example is an acquisition file. This list is only defined for SCIENCE raw file association. The format is the same as in 3.8.
Filled:	on creation
Qualifier:	RASSOC
Example:	RASSOC $DFO_RAW_DIR/2003-01-24/GIRAF.2003-01-25T05:49:21.036.fits ACQUISITION

3.10 Names of product files

Description:	This part contains all products of the AB. Apart from the qualifier, it has two fields. • If the data are CALIB data, the products are listed with their final name (calib_name). The second field checks whether the products have been ingested into the calibration archive. If yes, their calibration archive name is listed. • In case of SCIENCE data, reduced data are listed. The second field is always NONE. The directory is given in the comment line.
Filled:	on updating
Qualifier:	PRODUCTS
Examples:	for CALIB data: # product file(s) under $DFO_CAL_DIR/2003-01-24 (with status in

	calib archive) PRODUCTS GI_MBIA_030124A_1x1.fits M.2003-02-12T13:00:01.fits for SCIENCE data: # product file(s) under $DFO_RED_DIR/2003-01-24 (with status in calib archive) PRODUCTS r.GIRAF.2003-01-25T05:48:19.026_0000.fits NONE PRODUCTS r.GIRAF.2003-01-25T05:48:19.026_0001.fits NONE

3.11 Names of calibration products

Description:	This section has all associated calibration products that are needed for processing. There are four fields in addition to the qualifier: • A flag (VIRTUAL/REAL) for classifying if a calibration file is virtual or real. • Complete path and file name. Real files are listed with their final name and path. Virtual files are listed with their predicted name and the $DFS_PRODUCT path. • DO_CATG of the calibration file. • AB name associated with the calibration file. This name serves as link which is evaluated for packing. Reading this AB delivers the list of all raw calibration files that resulted in the calibration product. Virtual file names are verified upon updating (i.e., they are checked whether they have become real or not).
Filled:	on creation and on updating
Qualifier:	MCALIB
Example:	MCALIB VIRTUAL $DFS_PRODUCT/BIAS/2003-01-24/r.GIRAF.2003-01-25T01:05:18.771_tpl_0000.fits MASTER_BIAS GIRAF.2003-01-25T01:05:18.771_tpl.ab MCALIB REAL $DFO_CAL_DIR/2003-01-24/GI_MFLT_030124A_1x1.fits MASTER_FLT GIRAF.2003-01-25T10:17:19.121_tpl.ab

3.12 Associated calibration products

Description:	This section has additional associated calibration products that are needed for packing. It has the same format as 3.11. Only real calibration products are listed here (no virtual files).
Filled:	on creation and on updating
Qualifier:	MASSOC
Example:	MASSOC REAL $DFO_CAL_DIR/2003-01-24/GI_PDRS_030124A_1x1.tfits DRS_SETUP_TABLE GIRAF.2003-01-25T10:17:19.121_tpl.ab

3.13 Parameters

Description:	Some pipelines offer the option to append parameters to Reduction Blocks. These are listed here.
Filled:	on creation (if pipeline-supported)
Qualifier:	PARAM
Example:	PARAM 40

3.14 Associated QC1 parameters

Description:	On updating, a query to the qc1 database is made to look for associated QC1 parameters. Only those parameters are listed which went into the QC1 database.
Filled:	on updating

Qualifier:	QC1_PAR
Example:	QC1_PAR 53714.1132242 2003-01-24 23.112 15 443

3.15 Further associated information

Description:	Further (e.g. graphical) associated information is listed here (QC plots etc.).
Filled:	on updating (if pipeline-supported)
Qualifier:	FURTHER_PS| FURTHER_GIF | FURTHER_INI|FURTHER_PAF
Example:	FURTHER_PS $DFO_PLT_DIR/2003-01-24/r.GIRAF.2003-01-25T05:48:19.026_0000.fits.ps.gz FURTHER_GIF $DFO_PLT_DIR/2003-01-24/r.GIRAF.2003-01-25T05:48:19.026_0000.fits.gif

3.16 Status section

Description:	Each time an AB is created or updated, it receives a status line with a timestamp and a status message
Filled:	on creation and updating
Qualifier:	STATUS
Example:	STATUS: - created by 'find_assoc' on Fri Dec 20 17:47:04 CET 2002 by flames1 on dfo03 STATUS: - updated by 'update_assoc' on Fri Dec 20 17:47:49 CET 2002 by flames1 on dfo03

3.17 RB section

Description:	The complete RB information is contained in the AB. In this section, the RB information is made explicit. It follows RBS syntax.
Filled:	on creation and updating
Qualifier:	RB_CONTENT
Example:	RB_CONTENT recipe: giReduceScience RB_CONTENT RB_CONTENT instrument: giraffe RB_CONTENT RB_CONTENT $DFS_PRODUCT/SCIENCE/2003-01-24/r.GIRAF.2003-01-25T05:48:19.026 RB_CONTENT RB_CONTENT { RB_CONTENT $DFO_RAW_DIR/2003-01-24/GIRAF.2003-01-25T05:48:19.026.fits SCI_COMB_MOS RB_CONTENT } RB_CONTENT RB_CONTENT { RB_CONTENT $DFS_PRODUCT/BIAS/2003-01-24/r.GIRAF.2003-01-25T01:05:18.771_tpl_0000.fits MASTER_BIAS RB_CONTENT $DFS_PRODUCT/FLAT/2003-01-24/r.GIRAF.2003-01-25T01:33:04.822_tpl_0000.fits MASTER_FLAT RB_CONTENT $DFS_PRODUCT/WAVE/2003-01-24/r.GIRAF.2003-01-25T01:36:56.264_0000.tfits WAVE_COEF_TAB RB_CONTENT } RB_CONTENT

The complete example AB is found in the Appendix.

4 Association Blocks and databases

It seems both natural and useful to store all ABs in a database (AB database). This is useful when ABs are updated, and when they are read for preparing data packages. If inserted into, or linked to, the archive database, their association information could be used by archive tools.

E.g., no further association tools would then be needed to prepare data packages on-line. For pipeline-supported modes, reading ABs is superior to association from scratch since an AB carries information about successful RB execution.

To preserve historical association information, an option to store RBs (instead of, or in addition to, ABs) seems useful as well.

Appendix: Sample Association Block

Example is for GIRAFFE, MEDUSA (MOS) mode.

AB sections are coded:

information provided on creation
information provided on updating

```
# general information
TOOL_VERSION   1.0
CONFIG_VERSION giraffe_1.0.0

INSTRUMENT     giraffe
DATE           2003-01-24

DPR_CATG       SCIENCE
RAW_TYPE       SCIENCE
RECIPE         giReduceScience

AB_NAME        GIRAF.2003-01-25T05:48:19.026.ab
AB_EVENT       SINGLE
COMPLETENESS   COMPLETE

OBS_PROG_ID    60.A-9147(A)
OBS_ID         132137

MJD-OBS        52664.24188688
MJD-OBS_MODE   52664.24323892

# raw match key
RAW_MATCH_KEY  BIN=BIN1
RAW_MATCH_KEY  INS_SLIT_NAME=Medusa1
RAW_MATCH_KEY  INS_GRAT_ID=HR
RAW_MATCH_KEY  INS_GRAT_WLEN=WLN651.5

RBS_ROOT_NAME  r.GIRAF.2003-01-25T05:48:19.026

RB_NAME        GIRAF.2003-01-25T05:48:19.026.rX
LOG_NAME       GIRAF.2003-01-25T05:48:19.026.rblog

# raw file(s)
RAWFILE        $DFO_RAW_DIR/2003-01-24/GIRAF.2003-01-25T05:48:19.026.fits SCI_COMB_MOS

# associated raw file(s) (only for SCIENCE; taken from the same night only)
RASSOC         NONE

# product file(s) under $DFO_RED_DIR/2003-01-24 (with status in calib archive)
PRODUCTS       r.GIRAF.2003-01-25T05:48:19.026_0000.fits    NONE
PRODUCTS       r.GIRAF.2003-01-25T05:48:19.026_0001.fits    NONE
```

```
# associated mcalib file(s), used for processing
MCALIB  VIRTUAL $DFS_PRODUCT/BIAS/2003-01-24/r.GIRAF.2003-01-25T01:05:18.771_tpl_0000.fits MASTER_BIAS GIRAF.2003-01-25T01:05:18.771_tpl.ab
MCALIB  REAL    $DFO_CAL_DIR/2003-01-24/GI_MFLT_030124A_1x1.fits MASTER_FLT GIRAF.2003-01-25T10:17:19.121_tpl.ab

# associated mcalib file(s), used for packing (SCIENCE only)
MASSOC          MASSOC REAL $DFO_CAL_DIR/2003-01-24/GI_PDRS_030124A_1x1.tfits DRS_SETUP_TABLE GIRAF.2003-01-25T10:17:19.121_tpl.ab

# parameters for processing
PARAM           40

# associated QC1 parameters (QC1_TABLE = giraffe_flat):
QC1_PAR         53714.1132242 2003-01-24 23.112 15 443

# further associated information
FURTHER_PS      $DFO_PLT_DIR/2003-01-24/r.GIRAF.2003-01-25T05:48:19.026_0000.fits.ps.gz
FURTHER_GIF     $DFO_PLT_DIR/2003-01-24/r.GIRAF.2003-01-25T05:48:19.026_0000.fits.gif

STATUS          - created by 'findAssoc'      on Wed Feb 12 11:42:55 CET 2003 by flames1 on dfo03
STATUS          - updated by 'updateAssoc'    on Wed Feb 12 11:43:36 CET 2003 by flames1 on dfo03

# ============== RB section starts here ============================

RB_CONTENT      recipe: giReduceScience
RB_CONTENT
RB_CONTENT      instrument: giraffe
RB_CONTENT
RB_CONTENT      $DFS_PRODUCT/SCIENCE/2003-01-24/r.GIRAF.2003-01-25T05:48:19.026
RB_CONTENT
RB_CONTENT      {
RB_CONTENT      $DFO_RAW_DIR/2003-01-24/GIRAF.2003-01-25T05:48:19.026.fits SCI_COMB_MOS
RB_CONTENT      }
RB_CONTENT
RB_CONTENT      {
RB_CONTENT      $DFS_PRODUCT/BIAS/2003-01-24/r.GIRAF.2003-01-25T01:05:18.771_tpl_0000.fits MASTER_BIAS
RB_CONTENT      $DFS_PRODUCT/FLAT/2003-01-24/r.GIRAF.2003-01-25T01:33:04.822_tpl_0000.fits MASTER_FLAT
RB_CONTENT      $DFS_PRODUCT/WAVE/2003-01-24/r.GIRAF.2003-01-25T01:36:56.264_0000.tfits WAVE_COEF_TAB
RB_CONTENT      }
RB_CONTENT
RB_CONTENT
```