

DRAFT – FOR INTERNAL USAGE ONLY**Proposal for an Archive System for WFCAM/UKIDDS**

In order to support the archiving of data from the instrument WFCAM, the ESO Archive Facility will provide means to archive data remotely from Cambridge in the UK directly to Garching. Data will be archived into an NGAS Archiving Unit (NAU) at ESO HQ.

This document describes the scenario and in particular makes a specific proposal for how to handle the archiving and processing of this data.

The proposed system is generic in the sense that it could be re-used to handle archiving (remote or local) in other contexts. I.e., no specific features are 'hard-coded' for WFCAM.

System Description

The NGAS Archiving System at ESO HQ receiving the data from Cambridge, must be accessible externally to allow the remote archive client to connect. I.e., the network set-up at ESO HQ and at the site in Cambridge, must allow for such remote/external connections. The standard NGAS protocol based on HTTP is used.

The network bandwidth from Cambridge to Garching is approximately 1 MB/s. The expected amount of data produced from WFCAM is in the order of 100 GB/night, uncompressed. Since the data will be tile-compressed, the size of the frames will be smaller. A conservative estimate of the compression ratio is a factor 2, which gives us 50 GB/day (per 24 hours) to transfer and archive. With a sustained network transfer rate of 86 GB/day the bandwidth should not be a bottleneck. The NGAS system should be able to handle an estimated 400-450 GB/day of the given data type with the processing foreseen. Note, it is not foreseen to de-compress the data upon arrival, but certain checks on the data will be carried out, and features/parameters about each data file will be extracted in connection with the archiving.

NB: Should get uncompressed/tile-compressed file size to estimate better the handling time per frame.

Architecture

The online archive scenario required to handle the archiving of the WFCAM data, makes it necessary to define a new type of system compared to what is currently in used at the ESO La Silla and Paranal Observatories.

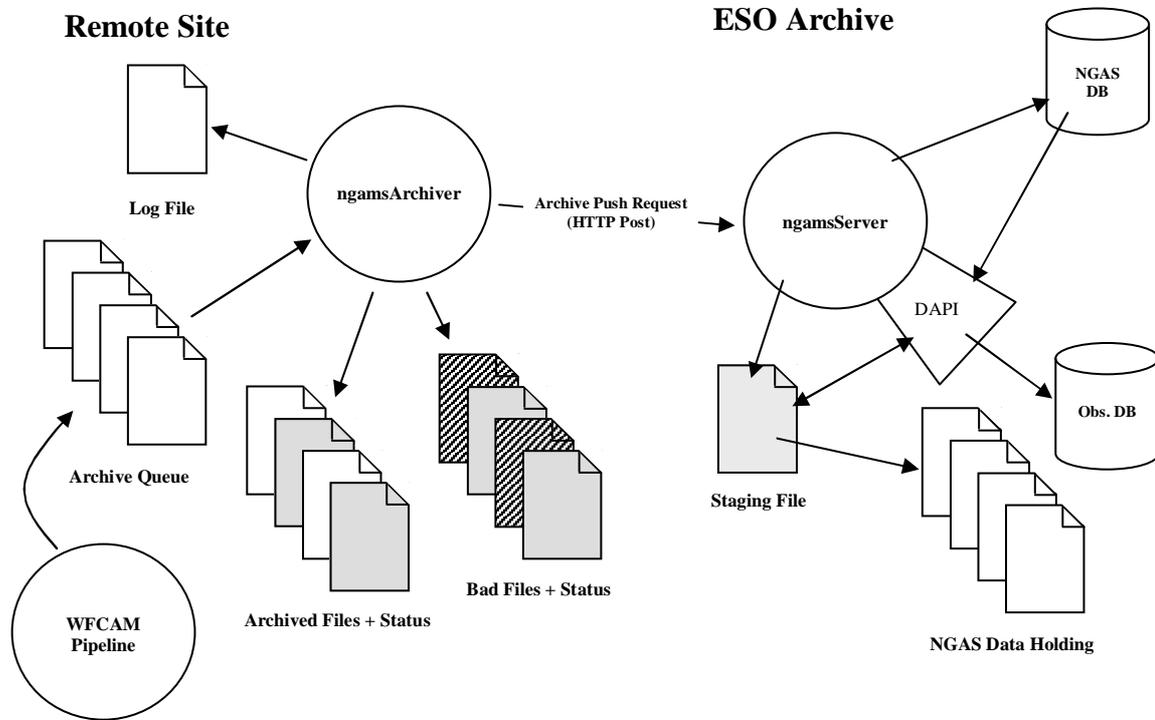
On the client side, no processing is to be performed. This means that the basic processing of the frames to make them 'ESO compliant' will have to be done at the server side. This consists of the following processing:

- Calculation + adding of the FITS checksum in the primary header (keyword: "HIERACH ESO CHECKSUM" - TBC).
- Adding the keywords "ORIGFILE" and "ARCFILE" in the header.
- Extraction of various keywords for the Observations DB.

Given the scenario, the most logical place to do this, is in the Data Archiving Plug-In (DAPI), which is executed by the NG/AMS Server to handle the specific processing and feature extraction of data files being archived. A new plug-in will be provided for this.

The only processing proposed on the client side could be the calculation of a checksum (CRC32, network byte order if relevant), which can be sent along with the HTTP request containing the data. This allows the server application at the ESO HQ to verify that the data has arrived correctly.

The following figure shows the complete scenario to handle the remote archiving of the WFCAM frames:



At the remote site the application providing the data (pipeline), will copy the data to be archived into the Archive Queue, which is a plain directory. The NG/AMS Archive Client polls periodically for new data in the Archive Queue. The data provider should create the file in the Archive Queue as a single operation, either by:

1. Creating a copy of the file elsewhere and move it to the Archive Queue.
2. Create a link pointing to the original copy of the file in the Archive Queue directory.

When a data file is found in the queue, the archive client tries to archive it into the remote NGAS Archiving Unit by means of an NG/AMS Archive Push Request (HTTP Post, data contained in body of request).

If the Archive Request is successful, the file (or link) will be moved from the Archive Queue to the Archived Files (directory). Along with the file/link itself, the XML status document from the remote NG/AMS Server will be created in the Archived Files Buffer. The name of the status document file will be: “<filename>-STATUS.xml”.

If the Archive Request fails, for instance if the file is bad, the file (or link) will be moved to the Bad Files Directory. The XML Status Document from NG/AMS will be stored in a file named “<filename>-STATUS.xml” in the Bad Files Directory. This XML document contains diagnostics about the problem encountered in connection with the file. This can be used for trouble-shooting purposes. A file is bad if e.g. keywords are missing in the header, or if the file is not a properly formatted FITS file for whatever reason. Usually

after a short time of operation, no files should be detected as 'bad'. Only if changes are introduced e.g. in the pipeline, such a problem may re-occur and should be corrected.

If the Archive Request fails for any other reason, the files stays in the Archive Queue, and it will be tried periodically to archive them. In particular, if the internal checksum calculated for the transfer of the frame from the remote site to the ESO HQ is bad, the file should be retransmitted.

The NG/AMS Archive Client can clean up the Archived Files Directory periodically. An expiration timeout can be specified for how long time the data should remain in the Archived Files Buffer. Before deleting a file, the NG/AMS Archive Client makes a cross-check against the data holding at ESO HQ to verify that the data has really been archived. This is done, based on the contents of the Archive Request Status Document, by issuing a File Access Request (“http://<NAU>:<Port No>/STATUS?file_access=<File ID>&file_version=<version>”) to the remote NG/AMS Server. Upon reception of this request, the NG/AMS Server will check if the file referenced has been registered in the NGAS DB and that the file is 'physically' available in the NGAS Archive Cluster at the ESO HQ. In this way it is ensured that no data file is deleted without having been archived properly into NGAS. Also the XML Status Documents will be removed (along with the corresponding FITS file).

A Log File is produced to be able to do trouble-shooting and to be able to make statistics in connection with the operation (transfer times etc.). The Log File will be rotated periodically if specified. I.e., a copy will be made, and a new Log File initiated.

At ESO HQ, the NG/AMS Server receives the file into the Staging Area on the Target Storage Media. Afterwards it invokes the Data Archive Plug-In (DAPI), which adds the keywords mentioned above and extracts the necessary keywords to be written in the Observations DB. Latter will be done 'table-driven' to make it possible to extract different keywords for different contexts, if it becomes necessary to set up a similar archive system for another instrument. It will also do a consistency check of the file. If a checksum is found in the HTTP Archive Request, the checksum of the file is checked. Also other features of the file is checked (TBD) and the parameters for NG/AMS generated.

NG/AMS Archive Client

The application running at the client side to archive the data into the NGAS system in Garching is named “ngamsArchiver”. It will be a deliverable of the NG/AMS package, and be part of the NG/AMS C-API sub-module. It is based on the NG/AMS C-API library functions.

The client should be as 'thin' as possible to make it easy and simple to use it on the client side. Also, the client should not produce any major requirements to the run-time environment on the client side.

ESO	Archiving System for WFCAM/UKIDDS 2003-11-18, J.Knudstrup, ESO/DMD
------------	---

The synopsis of the NG/AMS Archive Client is as follows:

```
% ngamsArchiver -host <host> -port <port> -fileQueue <dir>
                -archivedFiles <dir> -badFiles <dir>
                [-pollTime <poll time>]
                [-checksum] [-cleanUpTimeOut <timeout>]
                [-streams <no>] [-v <level>] [-logFile <filename>]
                [-logLevel <level>] [-logRotate <period>]
```

- whereby the command line parameters are:

-host <host>	Host ID (or IP address) of the Archiving Unit at ESO HQ.
-port <port>	Port number used by the NG/AMS Server at ESO HQ for serving clients.
-fileQueue <dir>	Directory in which files to be archived must be copied, or a link to the file created.
-archivedFiles <dir>	Directory containing the successfully archived files and files containing the NG/AMS XML Status Documents.
-badFiles <dir>	Directory containing Bad Files, which could not be archived.
[-pollTime <poll time>]	Time in seconds with which the NG/AMS Archive Client should poll the Archive Queue. Default value is 30s.
[-checksum]	If specified, a checksum (CRC32) will be generated and will be added in the HTTP headers (Content-disposition: ...; checksum=<checksum>). This is used at the server side to check that the file has been transferred correctly.
[-cleanUpTimeOut <timeout>]	Period of time (given in seconds) in which the successfully archived files in the Archived File Buffer should be kept.
[-streams <no>]	Number of parallel streams allowed. It will make sense to allow two streams to make it possible to process data on the server side, while the next file is being transferred. Default is 1 stream. If more is specified, the Archive Requests will be scheduled such that the next Archive Request is scheduled at the average time to handle an Archive Request divided by the number of streams allowed.
[-v <level>]	Verbose output. Default no output is given. A level in the range from 0 to 5 can be specified to produce more (or less) output.
[-logFile <filename>]	Name of Log File in which to produce log output.
[-logLevel <level>]	Level with which to log into the Log File. If not given, a level of 1 is the default value.
[-logRotate <period>]	Period of time (given in seconds) in which the Log File should be rotated.

ESO will deliver the NG/AMS Archive Client as a statically linked application for the target platform requested. I.e., there is no need for the external user to compile this application. It can be delivered on Solaris, HP-UX and Linux (tested primarily on RedHat, although other flavors of Linux would probably not be a problem).

Plan

The implementation plan suggested here, foresees two phases: 1) The first phase will provide a limited but operational system, 2) Implementation of (extended) features not implemented during Phase 1 and improvements/optimization.

The implementation is split into two phases to accommodate the relatively short termed schedule to support this.

The features to implement in each phase are described in the following sections.

Phase 1 - Features

The purpose of Phase 1 is to provide a system, which can already be tried out in practice to gain experience with the operational scenario.

The following features should be provided for Phase 1:

NG/AMS Archive Client:

The features provided for the Phase 1 NG/AMS Archive Client are:

- Polling of the Data Queue Directory.
- Calculation and adding of internal CRC32 checksum.
- Handling of Archive Requests including storing of NG/AMS XML Status files.
- Handling of re-transmission of failed Archive Requests (apart from Bad Files).
- Handling of Bad Files storing of NG/AMS XML Status files.
- Basic logging on stdout (Verbose Log) + in Log File.

WFCAM DAPI:

The features provided for the Phase 1 DAPI are:

- Verification of the internal NG/AMS checksum.
- Basic extraction of features for the NGAS DB + final processing of the request.
- Adding of keywords: "ORIGFILE", "ARCFIELD" and "HIEARCH ESO CHECKSUM".
- Extraction of selected keywords + writing of these in the Observations DB (basic implementation).
- Basic NGAS handling (already implemented).

Deadline: 2003-12-31 – TBD.

Cost: ~5 days.

Phase 2 - Features

The following features should be provided for Phase 2:

NG/AMS Archive Client:

The features provided for the Phase 2 NG/AMS Archive Client are:

- Cleaning up expired files from the Archived Files Buffer.

ESO	Archiving System for WFCAM/UKIDDS 2003-11-18, J.Knudstrup, ESO/DMD
------------	---

- Handling of parallel Archive Requests (thread based) + optimized scheduling based on handling time.
- Log Rotation.
- Improved/extended logging on stdout + in Log File.

WFCAM DAPI:

The features provided for the Phase 2 DAPI are:

- Extended test and verification of the FITS file if relevant/necessary.
- Generalization of the keyword extraction feature to make it 'table driven'.

Deadline: TBD.

Cost: ~5 days.

Phase 3 – Maintenance/Upgrade

After termination of Phase 2 it is not foreseen to provide new features as such unless something unexpected show up or the same archive system is to be used to archive data in another context, which has different requirements. The suggested implementation however, should be generic enough to handle most cases.

The NG/AMS Archive Client should be tested together with the new releases of NG/AMS and should be delivered to the remote site synchronized with the 'normal' releases of NG/AMS.