

SCUBA-2 DR Pipeline Project Office

University of British Columbia
 6224 Agricultural Road
 Vancouver, British Columbia
 CANADA
 V6T 1Z1

Tel: +1-604-822-2211

Fax: +1-604-822-5324

Email: jmolnar@ubc.ca


WWW: <http://scuba2.jach.hawaii.edu>

Document Title: Data Reduction Pipeline Algorithm Engine Interface

Document Number: SC2/SOF/IC210/03

Issue: 1.6

Date: 2005/05/24

Document Prepared By:	Tim Jenness	Signature and Date:	2005-05-23
Document Approved By:	B. D. Kelly	Signature and Date:	2003-11-28
Document Released By:	Janos Molnar	Signature and Date:	 2003-11-30



Contents

1 Introduction	3
2 Messaging Interface	3
3 File formats	3



1 Introduction

This document describes the interface between the SCUBA-2 Data Reduction Pipeline and the algorithm engines the Pipeline uses to perform the numeric processing. ORAC-DR provides pipelining infrastructure and recipes that control data processing but does not provide algorithms for actually performing the CPU-intensive data processing. This document describes how the algorithm engines (hereafter called “engines”) are controlled and how they retrieve the data.

2 Messaging Interface

Each engine has to be controllable by the Pipeline, but also has to support robust error handling so that the Pipeline system can determine if an error is fatal or can be worked around. This is ostensibly the reason why neither the Unix shell nor IRAF are supported by ORAC-DR. In addition to the basic robustness constraint, the only real constraint on the engine is that it uses a messaging system that is supported by Perl. Currently, the JAC has Perl experience with Starlink ADAM,^{8,3,1} AAO DRAMA^{2,5} and the industry-standard SOAP, with both Starlink ADAM and DRAMA currently available in ORAC-DR. The SCUBA-2 variant of ORAC-DR will use the following messaging systems:

- For existing Starlink applications, the ADAM interface will be retained. Whilst Starlink are considering moving their applications to SOAP we must make the assumption that they will not complete this work soon enough for our purposes. As long as ADAM is maintained on Unix there is no reason to switch at this time.
- For new applications written from scratch, DRAMA will be used (as specified in the original infrastructure proposal⁹).
- Some existing applications exist entirely in Java (e.g. the SPIRE FTS software). The FTS software team have decided to adopt DRAMA as their message system.

Req. ER8

The CASU WFCAM pipeline uses an alternate technique where the algorithm code (in C) is linked directly into the pipeline (using the perl XS interface⁷) removing the need for a messaging interface. SCUBA-2 will not be using this technique.

In ORAC-DR the messaging interface is only be used for passing control information between the Pipeline and the engine. Data is passed using the file names through the file system.

3 File formats

In principal, a remote engine can use any chosen data format, as long as routines exist to convert from the native format used by the Pipeline to the format required by the algorithm engine. Since, in practice, converting data formats every time a remote engine is called is extremely wasteful, it makes sense for the file format of choice to be the format supported by the majority of the applications. Since heavy use will be made of Starlink applications, the intermediate format used by the Pipeline will be NDF.¹⁰



Whilst, this is not part of the ICD, it is recommended that all new software should be written using an abstract interface to the underlying data file (i.e. a “data access layer”). For Java software it clearly makes sense to use the Starlink NDX/HDX classes, optimised for seamless I/O of FITS and NDF.⁴ A similar approach should be considered for the C implementation.

The raw data format is not important in this context since a single file conversion can usually be justified if necessary. The raw format is discussed in the DA/DR ICD.⁶

References

- [1] P. M. Allan. The ADAM software environment. In *ASP Conf. Ser. 25: Astronomical Data Analysis Software and Systems I*, pages 126–+, 1992.
- [2] J. A. Bailey, T. Farrell, and K. Shortridge. DRAMA: an environment for distributed instrumentation software. In *Proc. SPIE Vol. 2479, p. 62-68, Telescope Control Systems, Patrick T. Wallace; Ed.*, pages 62–68, June 1995.
- [3] A. J. Chipperfield. ADAM. Starlink User Note 144, Starlink Project, CCLRC, 2001.
- [4] D. Giaretta, M. Taylor, P. Draper, N. Gray, and B. McIlwrath. HDX Data Model: FITS, NDF and XML Implementation. In *ASP Conf. Ser. 295: Astronomical Data Analysis Software and Systems XII*, pages 221–+, 2003.
- [5] T. Jenness, F. Economou, R. P. J. Tilanus, C. Best, R. M. Prestage, P. Shimek, K. Glazebrook, and T. J. Farrell. Perl at the Joint Astronomy Centre. In *ASP Conf. Ser. 172: Astronomical Data Analysis Software and Systems VIII*, pages 494–+, 1999.
- [6] Tim Jenness. SCUBA-2 DA/DR interface control document. SCUBA-2 Project SC2/SOF/IC210/01, 2003.
- [7] Tim Jenness and Simon Cozens. *Extending and Embedding Perl*. Manning Publications, Inc., 2002. ISBN 1-930110-82-0.
- [8] M. D. Lawden and K. F. Hartley. ADAM – the starlink software environment. Starlink Guide 4, Starlink Project, CCLRC, 1992.
- [9] Nick P. Rees. Proposal for SCUBA-2 pipeline infrastructure. SCUBA-2 Project SC2/SOF/S200/011, 2003.
- [10] Rodney F. Warren-Smith. NDF – Routines for accessing the extensible N-Dimensional Data Format. Starlink User Note 33, Starlink Project, CCLRC, 1995.

