

420-WP-006-002

Establishing Science Software Exit Conditions for the Production Environment

White Paper

November 1996

White Paper--Not intended for formal review or
government approval.

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

<u>David Heroux /s/</u>	11/13/96
David Heroux, DPS Lead Engineer EOSDIS Core System Project	Date

SUBMITTED BY

<u>Karin E. Loya /s/</u>	11/13/96
Karin Loya, PDPS Manager EOSDIS Core System Project	Date

Hughes Information Technology Systems
Upper Marlboro, Maryland

This page intentionally left blank.

Abstract

This white paper addresses the outstanding issues resulting from the original EOS Core System (ECS) response to a Preliminary Design Review (PDR) Review Item Discrepancy (RID) 423. Those issues concern the exception conditions which can arise during a Product Generation Executive (PGE) run and the manner in which the ECS system can assimilate and carry out the terms of those conditions. A standard list of conditions is defined along with a template for additional, user-defined, conditions.

Keywords: condition category, exit condition, pge dependency

This page intentionally left blank.

Contents

Abstract

Contents

1. Introduction

1.1 Purpose.....	1-1
1.2 Organization.....	1-1
1.3 Review and Approval.....	1-1
1.4 References.....	1-2

2. Context - ECS Production System

2.1 Overview.....	2-1
2.2 Background.....	2-1
2.3 Release A Focus.....	2-2
2.4 Flow of Control.....	2-3
2.4.1 Production Control.....	2-3
2.4.2 Operator Control.....	2-3
2.5 Condition Categories.....	2-3

3. Communication Mechanism

3.1 Pre-defined Condition Set.....	3-1
3.1.1 Allocation Mix.....	3-1
3.1.2 Sets of Conditions.....	3-2

4. Responsibilities

4.1 SSI&T Role.....	4-1
4.2 Developer's Direction.....	4-1
4.2.1 Custom PGE Conditions	4-2
4.2.2 PGE Dependency Rules.....	4-2
4.2.3 PGE Termination	4-4
4.2.4 Providing Information to the DAAC	4-4

Tables

1-1.	White Paper to CDRL Migration	1-2
2-1.	Exit Condition Categories	2-4
3-1.	Allocation of Unix Exit Codes	3-2
3-2.	Base Set of PGE Exit Conditions.....	3-2
4-1.	PGE Dependency Worksheet.....	4-3

Abbreviations and Acronyms

1. Introduction

1.1 Purpose

The purpose of this document is to address the need for providing a communication pathway from the science software to the ECS production system. Defining one that can be implemented in a Rel. A timeframe, and that can be expanded to accommodate future science software releases, is also a goal of this white paper.

The mechanics of establishing this communication link are defined, along with a set of predefined messages that the science software can use to "talk" to the production system. To convey science-specific messages to the production system, a user-customizable template is provided.

This white paper derives its origin from PDR RID 423¹, which addressed the concern regarding various exception conditions and their effect on the system.

1.2 Organization

This paper is organized as follows:

Section 1 of this document defines the purpose and review policy for the definitions defined herein.

Section 2 of this document provides an overview and some background on the ECS production system to provide the context. Some constraints on the usage of the proposed mechanism, for the Release A timeframe, are presented here as well.

Section 3 of this document defines exit conditions and discusses how control is passed from the science software to the production system; the role of the production operator, in this regard, is also briefly discussed

Section 4 of this document provides some direction for the developer and integration team looking to establish exit conditions for a PGE.

1.3 Review and Approval

This White Paper is an informal document approved at the Office Manager level. It does not require formal Government review or approval; however, it is submitted with the intent that review and comments will be forthcoming. All comments on the pre-defined exit conditions are due to the author no later than December 13, 1996; user-definable exit conditions/dependencies are submitted along with the Delivered Algorithm Package (DAP) for each PGE.

¹ <http://edhs1.gsfc.nasa.gov/Info/pdr/sdps-rids-toc.html>

The ideas expressed in this White Paper are valid for the August 1996 to January 1997 period; the concepts presented here are expected to migrate into the following formal CDRL deliveries:

Table 1-1. White Paper to CDRL Migration

White Paper Section	CDRL DID/Document Number
2, 3 & 4	305-CD-011-002

Questions regarding technical information contained within this Paper should be addressed to the following ECS and/or Goddard Space Flight Center (GSFC) contacts:

- ECS Contacts
 - David P. Heroux, DPS Lead Engineer, 301.925.0753, dheroux@eos.hitc.com

Questions concerning distribution or control of this document should be addressed to:

Data Management Office
The ECS Project Office
Hughes Information Technology Corporation
1616 McCormick Drive
Landover, MD 20785

1.4 References

333-CD-003-004

Release A SCF Toolkit Users Guide for the ECS Project

2. Context - ECS Production System

2.1 Overview

An important aspect of executing science software in the DAAC processing environment, will be in the ability of the Data Processing Subsystem (DPS) to capture the final state of science processing. To this end, it will be necessary to establish a communication pathway from the science software to the production system. Given the current capabilities of the DPS design, the obvious mechanism for satisfying this requirement is through the prudent use of the Portable Operating System Interface (POSIX) termination feature². By employing this feature, exit conditions can be neatly propagated to the production system. But while an exit condition of success or failure allows for simple binary branching within a production tree, this does not provide adequately for the more complex decision logic which will be necessary to ensure that a production schedule can be met to the fullest extent possible. That is, it is not enough to have a PGE fail and terminate processing along that branch. Rather we must be able to ascertain why a PGE failed so that we may possibly continue execution along the same branch, after having taken measures to correct the deficit which initiated the failure of the PGE. In order to accomplish this, all science software (PGEs) will need to establish an exit condition which is relevant to the DPS in this regard, and relevant to downstream science processing for all other production concerns. The obvious way to encourage this is to publish a list of such exit conditions. What is not obvious however, is the complete set of such conditions. This paper proposes a base set of science software exit conditions, along with a procedure for expanding the base set; refinement of this set of exit conditions may occur through the normal review process.

2.2 Background

The DPS design integrates a Commercial Off-the-Shelf (COTS) scheduling engine (AutoSys) with custom software for, among other things, monitoring the exit status of science software PGEs. To facilitate the execution of PGEs within this production environment, "jobs" are defined, in terms of the COTS syntax, to enforce the data dependency conditions which drive a production stream. In the absence of any error condition, the production stream proceeds to completion without any intervention on the part of the DPS, or the Production Operator. Currently (IR-1), if an error condition occurs during PGE processing, the "DPS" (essentially AutoSys) can only halt the production stream. However, this is not a limitation on the production software. Rather, it is simply a limitation on the number of valid exit conditions which are currently recognized by the system. Given a reasonable range of exit conditions, the DPS should be capable of: diagnosing certain hardware conditions which may result in the re-hosting and re-

² POSIX defines an exit routine for C, Fortran and shell script languages.

execution of a PGE³, recognizing some software conditions which will trigger a context-based alarm to be sent to the Production Operator⁴, disabling selected downstream processing, or enabling alternate downstream processing to the extent that this concept is supported by the Production Planning Subsystem.

2.3 Release A Focus

The intent at Rel. A is to provide an additional level of control that is not currently possible in IR-1. To yield control to the DPS where this makes sense, and to put control into the hands of the capable Production Operator where such intervention can prove invaluable. However, in Rel. A, the extent to which the DPS can control the production stream is governed by several factors.

Chief among them is the user's willingness to take advantage of this capability. Again, unless more than the simple binary exit condition is used in the PGE, the DPS can only realize a "success", or generally "failed" condition; for the latter condition, the only response can be to shunt all downstream processing. Other factors include the degree to which PGEs depend on one another, and how often production is re-planned. The more dependencies that exist between PGEs, the more DPS can be applied to carry out user prescribed exit-condition-based activations and deactivations; presumably, if certain knowledge could be propagated to dependent PGEs, this could form the basis for deciding whether, or not, a dependent PGE should even be started. In so doing, DPS can actually preserve resources that would otherwise be consumed by "affected" PGEs (i.e. those for which an attempt at execution would prove to be fruitless). The belief here is that computing resources are best allocated to those PGEs whose outcomes are more promising than others. The net result is that the Production Operator's time is not spent performing unnecessary recovery procedures. More importantly, the planned events of the day become that much more achievable. However, it should be noted that this feature does not work across production plan boundaries, since PGE dependency information does not extend from one plan to the next. For this reason, dependent DPRs which have not started prior to a re-plan, will lose their dependency information once a new plan is activated. New plan activations have the affect of removing completed DPR information from the AutoSys database.

In general, since the activation of alternate DPRs is not a supported feature of Rel. A Production Planning, the mechanism proposed in this paper should be used as a means of achieving efficient recovery (where possible) of PGEs, diagnostic processing of PGEs, and more effective use of production resources.

³ e.g., a disk device fault manifests itself as a software error reported by a file I/O tool. An exit condition which appropriately represents this error, may be sufficient to determine an appropriate response to the condition, possibly without operator intervention.

⁴ e.g., an exit condition which indicates that a software error has occurred, due to an ill-defined runtime parameter, may be conveyed to an operator through the Alarm Manager utility. If so directed, the operator may be able to take the necessary corrective measures and re-execute the PGE.

2.4 Flow of Control

Depending on the nature of the PGE exit condition, current, or downstream processing can be controlled by either the Planning and Data Processing System (PDPS), or the Production Operator.

2.4.1 Production Control

A goal of the DPS design is to provide for as much automation as possible in the generation of science products. To support this, the science software needs to identify exit conditions, by returning exit codes, that are handled automatically by the Processing Subsystem. Such exit conditions include: File I/O problems, Memory access problems, Product Quality conditions, Instrument specific conditions, Science Software conditions and naturally a successful exit condition. Note however that the degree of automation varies according to the type of condition. Refer to table 2-1 for a list of condition categories and the DPS actions applicable to each.

2.4.2 Operator Control

For those exit conditions which can not be handled by the DPS alone, a means of soliciting assistance from the Production Operator will be required. Given the current capabilities of the COTS scheduling software, the most promising method for achieving this is through the use of the application's "Alarm Management" utility. This interface permits the DPS to selectively channel messages to the operator's "Alarm Management" window. These pre-defined messages should correspond to exit conditions that warrant the operator's attention. So when a condition such as this occurs, the DPS will retrieve the appropriate message and trigger an "alarm" event. This action will populate the operator's view with enough information to begin an investigation into the cause of the condition.

Some exit conditions which may be candidates for operator intervention include: Toolkit setup problems, Toolkit termination problems, science software parameter anomalies, science data file anomalies, and problems with software library services.

It is important to note that the accuracy of the forecast product availability, per the Production Plan, will be affected by those PGE exit conditions which do not permit the production stream to continue without operator assistance.

2.5 Condition Categories

The following table categorizes PGE exit conditions based on the type of exception and the possible subsystem reaction to the condition.

Table 2-1. Exit Condition Categories

Condition Category	Exception Type*	Subsystem Action
--------------------	-----------------	------------------

automated recovery

RESOURCE	HF/SF/SE	Possible auto/manual restart if resources permit
----------	----------	--

redirect/shunt production flow

SCIENCE	SF/SE	Success or failure determine subsequent processing; Science debug activation
QUALITY	QA	Product quality determines subsequent production
INSTRUMENT	IE	Instrument state determines subsequent production

manual intervention

SETUP	SE	Alert operator; Toolkit setup problem
PARAMETER	SE	Alert operator; runtime parameter anomaly
DATA	SE	Alert operator; data file anomaly
LIBRARY	SF/SE	Alert operator; COTS software problem; Toolkit debug activation

- * HF : Hardware Fault - detected by DPS; potentially indicated by SF or SE.
- SF : Software Fault - detected and trapped by science software.
- SE : Software Error - detected by science software/Toolkit.
- IE : Instrument Error - detected by science software.
- QA : Quality Assurance - detected by science software

3. Communication Mechanism

3.1 Pre-defined Condition Set

Founded on the understanding of the production environment, and the diversity of software that is expected to operate within that environment, an initial, base set of PGE exit conditions has been compiled. This set is presented in Table 3-2. It is expected that some changes will need to be made as our understanding of the science software and its interaction with the production environment become better understood. Therefore, we should not consider this to be the final set. Rather, consider this to be a working set. Science teams should use this list as an aid, while assessing the needs of their software in this matter. Since these codes define the only communication link between the science software and the production environment, it would be beneficial for each team to employ them where it makes sense to. Whereas some teams will only require their software to report a condition of success or failure, other teams may require more extensive conditions to fully manage the flow of their production software.

To meet the needs of all teams and provide a level of continuity for the management of these exit conditions, a statement formalizing these and/or other conditions will be made on the ECS Data Handling System (EDHS) home page⁵ at the conclusion of the review cycle for this paper.

3.1.1 Allocation Mix

While there is recognition of the need to support additional exit conditions over time, the total number of conditions is limited by the standards imposed on the current generation of Unix operating systems⁶; additional limitations exist due to the number of conditions which represent valid Unix errors and those already reserved by the Toolkit. Given these limitations, the number of error codes which can safely be employed for our purpose is a mere 40. While this number may seem overtly limiting, in fact we may be hard-pressed to define that many exit conditions which allow for DPS/Operator intervention, or affect the activation of processing downstream. Below, Table 3-1 reveals how Unix exit codes are interpreted by the DPS.

Of particular interest to the science software developers, the exit codes presented in bold provide the means to direct the course of downstream processing; all other codes represent potential termination events.

⁵ <http://edhs1.gsfc.nasa.gov>

⁶ This limitation does not extend to the Autosys Scheduling software to the same degree.

Table 3-1. Allocation of Unix Exit Codes

Shell Status Code (inclusive ranges)	Reserved Function	AutoSys Perspective	ECS DPS* Perspective
0	Unix/PGE	Success	Hot
1 - 199	Unix	Failure	Frozen
200-202	PGE	Failure	Cold
203-222	PGE	Success	Hot
223-239	PGE	Failure	Warm
240-255	Toolkit	Failure	Warm

* The relative "temperature" associations are only used to describe how the ECS system (specifically the DPS) interprets the various status code values:

Frozen - not interpreted

(C) cold - unrecoverable fault, or major error ... processing halted

(W) warm - potentially recoverable fault, or correctable error ...
intervention required

(H) hot - success, or with minor error ... proceed uninterrupted

3.1.2 Sets of Conditions

Table 3-2. Base Set of PGE Exit Conditions

Condition Category			
Exit Condition	Unix code	T*	Message

RESOURCE			
ECS_SHMEM	223	W	Shared memory access error
ECS_DEVICE	224	W	Disk device access error
ECS_PERMISSION	225	W	Execute/Access permission problem
ECS_DATABASE	226	W	Database access error
ECS_NETWORK	227	W	Network access error
RESERVED	228	W	

SCIENCE			
ECS_MEMORY	200	C	Memory fault
ECS_FLOAT	201	C	Floating point exception
RESERVED	202	C	Future use
ECS_DEBUG	234	W	Debug Trigger
	203	H	
	204	H	
	205	H	
	206	H	
	207	H	

- n.b. If a PGE exits with condition ECS_DEBUG, then a Toolkit runtime parameter will be enabled and the PGE will be initialized for restart. (reference logical id = 10911). If debug has been enabled, this parameter will be set to "1", otherwise, it will be set to "0". It shall be left to the developer to interpret the meaning of this parameter. In general though, it is provided as a means for the PGE to signal that a restart should occur, perhaps to gather additional information during a second run. In Rel. A, the actual restarting of the job is left to the operator's discretion (i.e. it is not automatically restarted).

QUALITY			
	208	H	
	209	H	
	210	H	
	211	H	
	212	H	

INSTRUMENT			
	213	H	
	214	H	
	215	H	
	216	H	
TRMM_ATT_REPAIR	217	H	Trigger attitude gap-filler PGE to fire

Example: This exit condition may be used by the DPS DPREP PGE in order to activate an additional "pre-planned" PGE when the primary PGE has detected gaps in the FDF provided attitude data.

SETUP			
RESERVED	240	W	Future use
RESERVED	241	W	Future use
PGS_SH_SMF_MSSLOGFILE	242	W	Event Logging has been disabled (IR1 ONLY)
PGS_SH_PC_TRUNC	243	W	Element(s) accessed from the PCF have been truncated due to size limitations
PGS_SH_PC_TOOL_ERR	244	W	Non-specific error occurred during the last PCF access
PGS_SH_PC_NODATA	245	W	Expected data was not located during the last access of the PCF
PGS_SH_SYS_PARAM	246	W	Insufficient arguments passed to command
PGS_SH_MEM_INIT	247	W	Initialization of shared memory failed
PGS_SH_PC_DELETETMP	248	W	Error detected during the deletion of temporary scratch files (ONLY applies if shared memory not used)
PGS_SH_SMF_SEND_RUNTIME	249	W	Runtime transmission is disabled, or faulty
PGS_SH_SMF_SEND_LOGFILE	250	W	Logfile transmission is faulty
PGS_SH_MEM_TERM	251	W	Cleanup of shared memory failed
PGS_SH_SMF_LOGFILE	252	W	One or more of the Toolkit log files could not be opened
PGS_SH_PC_LOADDATA	253	W	PCF not properly loaded into shared memory
PGS_SH_PC_ENV	254	W	Bad Process Control environment
PGS_SH_SMF_SHMMEM	255	W	Shared memory has not been initialized; ready to proceed without it.

PARAMETER			
ECS_BAD_PARM	229	W	Bad parameter value detected
ECS_MISSING_PARM	230	W	Insufficient number of parameters
RESERVED	231	W	Future use
RESERVED	218	H	Future use
ECS_SUSPECT_PARM	219	H	Suspect parameter used

DATA			
ECS_FILE_ACCESS	232	W	File access problem detected
ECS_MISSING_DATA	233	W	Insufficient number of input files
ECS_SUSPECT_META	220	H	Suspect file metadata written
ECS_SUSPECT_DATA	221	H	Suspect file contents written
ECS_PARTIAL_DATA	222	H	Incomplete product generated

LIBRARY			
ECS_TOOLKIT	235	W	Unhandled Toolkit error; trip debug
ECS_IMSL	236	W	Unhandled IMSL error
ECS_ODL	237	W	Unhandled ODL error
ECS_HDF	238	W	Unhandled HDF/HDF-EOS error
RESERVED	239	W	Future use

This page intentionally left blank.

4. Responsibilities

4.1 SSI&T Role

Having a well-defined set of exit conditions to incorporate into the science software interface is the essential first step. The next step is to define a mechanism for establishing another job's dependency on these exit conditions. In order for the policies implied by these exit conditions to be carried-out, the "jobs" which represent the science software, within the COTS scheduling application, must embed the dependency information within their definition. Since all "job" definitions are constructed from the information contained in the PGE profile database, this dependency information will need to be captured during the Science Software Integration and Test process, as this is when the PGE profile database gets populated. Fortunately, the syntax for defining dependency information within the COTS package is straight forward. What is required for each PGE is a list of the exit conditions on which it depends, along with the identification of the "up-stream" PGE and the conditional operators which apply to each exit condition (n.b., this feature only extends to the so called "Hot" exit conditions; all others result in the automatic stoppage of the affected production flow). Ideally, the syntax guiding the formation of such information in the PGE Profile database, should closely match that which is used when presenting this information to the COTS application.

Pursuant to this, the design of the PGE profile database has been modified to directly support the capture of all relevant PGE exit information. While the source of such information is the sole responsibility of the science software development organization, its migration into the PDPS database depends entirely on the Science Software Integration and Test (SSI&T) Operator. To assist in the matter, an exit condition template will be presented, in the next section, with the goal of facilitating the transfer of all key exit information⁷.

4.2 Developer's Direction

In the absence of any action on the part of the instrument development team, the only exit conditions that a PGE can realize are those belonging to the set of Toolkit Setup conditions. Therefore, in order for a PGE to capitalize on this Production feature, some minimal effort must be extended on the part of the developers. The scope of such effort is generally confined to the exception handling blocks of scripts and executables (the lack of which usually signifies a perilous endeavor).

The following sections detail the steps that must be taken by the developer for each delivered PGE.

⁷ In order to effect the transfer, developer provided information must be transcribed into the PGE ODL template by means of a conventional editor; the latter template being a by-product of the SSI&T process.

4.2.1 Custom PGE Conditions

□ For non failure conditions, customize a set of exit codes, and their associated message text (keep messages to less than 60 characters for Operator readability). Fill-in up to 15 exit condition records providing: mnemonic, code and message, in Table 3-2 ("condition category" is not significant at the present time).

- 15 codes in the range [203-217] have been reserved for this purpose. They have been divided up across 3 condition categories (science, quality and instrument), to facilitate exit conditions that may become standardized in the future by the science community; the implication being that standardized conditions initiate a standardized response on the part of the Processing subsystem (e.g., do not proceed with product archival if the quality condition is x). Until such time, the user can define any of these codes in a manner of their choosing, with the understanding that none shall be interpreted by the DPS as a failure event for the exiting PGE.
- This allows each PGE to have its own unique set of exit conditions.
- Custom PGE messages will be used in the generation of Processing Reports.
- More importantly, this information can be used to formulate conditional rules to govern downstream processing of "planned" data processing requests (DPRs).

n.b It is important not to confuse the concept of exit condition rules with that of production rules (the former being a mechanism for all releases, the latter being a Release B. feature).

4.2.2 PGE Dependency Rules

□ Define a set of 1 or more dependency rules which are based on the exit conditions of other PGE(s). Developers can use this worksheet in Table 4-1 to define explicit activation conditions based on the "hot" exit conditions of up-stream PGEs.

4.2.3 PGE Termination

□ Unless there is an overriding failure condition, exit with one of these PGE-specific condition codes.

- This will satisfy any exit dependencies that may have been established for downstream PGEs.

□ Communicate the presence of a failure condition to the Production Operator, by using one of the ECS defined condition codes.

- This allows for the possibility of a restart, once corrective measures have been taken by the Production Operator.

n.b. To facilitate possible restart situations, always open product output files with 'write' mode (ref. Table 6-7 of SCF Toolkit Users Guide for the proper mode value).

- This also allows Processing Reports to detail the exit condition present at the time of failure.

4.2.4 Providing Information to the DAAC

□ Create a single document from the 2 tables and incorporate it into the Delivered Algorithm Package (DAP).

Abbreviations and Acronyms

COTS	Commercial Off-the-shelf
DAAC	Distributed Active Archive Center
DAP	Delivered Algorithm Package
DPR	Data Processing Request
DPREP	Data Pre-Processing
DPS	Data Processing Subsystem
ECS	EOSDIS Core System
EDHS	ECS Data Handling System
FDF	Flight Dynamics Facility
IR-1	Interim Release 1
ODL	Object Description Language
PDPS	Planning and Data Processing System
PGE	Product Generation Executive
POSIX	Portable Operating System Interface
QA	Quality Assurance
RID	Review Item Discrepancy
SCF	Science Computing Facility
SSI&T	Science Software Integration and Test
SSW	Science Software
SW	Software