Clouds and the Earth's Radiant Energy System (CERES)

Data Management System

Operator's Manual

CERES AuTomAted job Loading sYSTem (CATALYST)

Version 6

Primary Authors

T. Nelson Hillyer, Joshua C. Wilkins

Science Systems and Applications Incorporated (SSAI)
One Enterprise Parkway, Suite 200
Hampton, VA 23666

NASA Langley Research Center
Climate Science Branch
Science Directorate
21 Langley Boulevard
Hampton, VA 23681-2199

SW Delivered to CM: July 2014
Document Date: July 2014

Document Revision Record

The Document Revision Record contains information pertaining to approved document changes. The table lists the date the Software Configuration Change Request (SCCR) was approved, the Version Number, the SCCR number, a short description of the revision, and the revised sections. The document authors are listed on the cover. The Head of the CERES Data Management Team approves or disapproves the requested changes based on recommendations of the Configuration Control Board.

Document Revision Record

SCCR Approval Date	Version Number	SCCR Number	Description of Revision	Section(s) Affected
9/19/2012	V1	931	• Initial version of document.	All
			• Corrected definition of ASDC. (12/11/2012)	All
11/14/2012	V2	937	• Updated for CATALYST build 1.	All
			• Updated for CATALYST build 1.0.3. (08/27/2013)	All
			• Added job lifecycle description and caveats section and extended the CATALYST_WONT_RUN description. (09/12/2013)	Secs. 1.9, 1.10, and 1.12
			• Added warning information, error condition handling, and added example information to "Best Practices" section. (09/24/2013)	Secs. 1.4.1, 1.8, 1.10, & 1.11
			• Fixed some cross reference link issues. (10/10/2013)	All
			• Referencing CATALYST Test Plan for discussion of installation steps instead of duplicating it in this document. (11/19/2013)	Sec. 1.4.1
			• Added table listing the possible server configuration parameters. (11/19/2013)	Sec. 1.3
			• Added sections describing methods to start the CATALYST server. (11/19/2013)	Sec. 1.4.2
			• Added new log file reference in table. (11/19/2013)	Sec. 1.8
			• Edited the Operators Console descriptions and screenshots to reflect the changes made to it. Added several new menu options. (11/19/2013)	Sec. 2.1
			• Added details to installation description. (12/18/2013)	Sec. 1.4.1.1
			• Added description and recovery information for loss of database connectivity. (12/18/2013)	Sec. 1.13.2

Document Revision Record

SCCR Approval Date	Version Number	SCCR Number	Description of Revision	Section(s) Affected
11/14/2012 (Cont'd)	V2	937	• Added recovery procedures for Operator's Console errors. (12/18/2013) • Added installation package file content listing. (12/18/2013) • Added database production availability section. (12/18/2013) • Added description assigning multiple recipients to PR deletion notifications and logging database alarm emails to the CATALYST configuration file. (02/20/2014)	Sec. 2.3 App. D App. E Sec. 1.3
02/06/2014	V3	996	• Added database patch instructions and updated database content listing.	Sec. 1.4.1.1.4 & App. E
04/16/2014	V4	1008	• Added description of improved CATALYST log files. • Added explicit listing of users with read-only privileges. • Added description of patch installer functionality. • Added LogUtils.pm and JIRA ticket tests to CATALYST directory and file listing.	Sec. 1.8 Sec. 1.4.4 Sec. 1.4.1.1.4 App. D
06/14/2014	V5	1021	• Updated description of permissions to reflect change allowing users marked as 'operator' to submit PRs to CATALYST. • Added listing for the ASDC epilogue output log location.	Sec. 1.4.4 Sec. 1.8
07/24/2014	V6	1031	• Updated the layout description of the Operator's Console to reflect the additional preferences and slight visual change to the Job Table. • Added more details related to the preferences dialog box. (08/12/2014)	Sec. 2.1 Sec. 2.1

Preface

The Clouds and the Earth's Radiant Energy System (CERES) Data Management System (DMS) supports the data processing needs of the CERES Science Team research to increase understanding of the Earth's climate and radiant environment. The CERES Data Management Team works with the CERES Science Team to develop the software necessary to support the science algorithms. This software, being developed to operate at the Langley Atmospheric Science Data Center (ASDC), produces an extensive set of science data products.

The DMS consists of 12 subsystems; each subsystem contains one or more Product Generation Executables (PGEs). Each subsystem executes when all of its required input data sets are available and produces one or more archival science products.

This Operator's Manual is written for the data processing operations staff at the Langley ASDC by the Data Management Team Systems group who are responsible for the CATALYST system. This document describes the CATALYST software, and outlines installation, execution, and monitoring procedures. In addition, all CATALYST error messages and subsequent actions required by the ASDC operations staff are included.

Acknowledgment is given to the CERES Documentation Team for their support in preparing this document.

TABLE OF CONTENTS

Section Page

Document Revision Record ii

Preface iv

1.0 CATALYST Server 3

1.1 Server Details 3

1.1.1 Responsible Persons 3

1.1.2 E-mail Distribution List 3

1.2 Operating Environment 3

1.2.1 Runtime Parameters 3

1.2.2 Environment Script Requirements 3

1.2.3 Execution Frequency 4

1.2.4 Memory/Disk Space/Time Requirements 4

1.2.5 PGE Handlers 4

1.3 Server Configuration Parameters 5

1.4 Operating Procedures 7

1.4.1 Installing the CATALYST Server Software 7

1.4.1.1 Installation 7

1.4.1.1.1 Installation Instructions 7

1.4.1.1.2 Installation Description 7

1.4.1.1.3 Determining if the Installation is Successful 7

1.4.1.1.4 Installing CATALYST Server Updates 7

1.4.1.1.5 Installing Logging Database Patch Scripts 8

1.4.2 How to Start the CATALYST Server 8

1.4.2.1 Interactive 8

1.4.2.2 Detached/Daemonized 9

1.4.2.3 Linux System Service (init.d entry) 9

TABLE OF CONTENTS

Section Page

1.4.3 How to Verify the CATALYST Server is Running 9

1.4.4 Editing the CATALYST Server's Access Control List 10

1.4.5 Terminating the CATALYST Server 11

1.5 Processing Request Submission from the Command Line 11

1.5.1 Creating a PR Load Script 11

1.6 Expected PR Instantiation Time 11

1.7 Execution Evaluation 12

1.7.1 Exit Codes 12

1.8 Operational Log Files/Directories. 12

1.9 Operational Files/Databases. 13

1.10 CATALYST Job Life Cycle 13

1.10.1 Job States 13

1.10.2 Manual Actions Influencing State Transitions 14

1.10.3 Graphical Workflow 14

1.11 Automatic Determination of Will Not Run Conditions 15

1.12 Best Practices 17

1.13 Caveats 17

1.13.1 PGE Execution 17

1.13.1.1 Periodic JSS_ERRORs during Delayed Execution of Inversion CER4.5-6.1P4/5 17

1.13.2 Loss of Database Connectivity 19

2.0 Operator's Console 20

2.1 Layout and Features 20

2.2 Obtaining the Operator's Console Software 23

TABLE OF CONTENTS

Section Page

2.3 Recovering from a Operator's Console Failure 24

Appendix A - Acronyms and Abbreviations A-1

Appendix B - Error Messages for CATALYST B-1

Appendix C - CATALYST Environment C-1

Appendix D - List of Files Created On First Install D-1

Appendix E - Product Availability in CATALYST Database E-1

E.1 CATALYST 1.0.4 Installation Package + Logging Database Patch (SCCR 996) E-1

LIST OF FIGURES

Figure Page

Figure 1-1. Flow Mapping for the CATALYST Server 5

LIST OF TABLES

Table Page

Table 1-1. Subsystem Software Analysts Contacts 3

Table 1-2. Runtime Parameters for Server 3

Table 1-3. Memory/Disk Space/Time Requirements for CATALYST Server 4

Table 1-4. PGEs Handlers for the CATALYST Server 4

Table 1-5. Configuration Parameters 5

Table 1-6. CATALYST Privilege Categories 10

Table 1-7. Exit Codes for CATALYST Server 12

Table 1-8. Temporary Files Listing for CATALYST Server 12

Table 1-9. Operational File Listing for CATALYST Server 13

Table 2-1. Dependencies 20

Introduction

CERES is a key component of EOS and NPP. The first CERES instrument (PFM) flew on TRMM, four instruments are currently operating on the EOS Terra (FM1 and FM2) and Aqua (FM3 and FM4) platforms, and NPP (FM5) platform. CERES measures radiances in three broadband channels: a shortwave channel (0.3 - 5 μm), a total channel (0.3 - 200 μm), and an infrared window channel (8 - 12 μm). The last data processed from the PFM instrument aboard TRMM was March 2000; no additional data are expected. Until June 2005, one instrument on each EOS platform operated in a fixed azimuth scanning mode and the other operated in a rotating azimuth scanning mode; now all are typically operating in the fixed azimuth scanning mode. The NPP platform carries the FM5 instrument, which operates in the fixed azimuth scanning mode though it has the capability to operate in a rotating azimuth scanning mode.

CERES climate data records involve an unprecedented level of data fusion: CERES measurements are combined with imager data (e.g., MODIS on Terra and Aqua, VIIRS on NPP), 4-D weather assimilation data, microwave sea-ice observations, and measurements from five geostationary satellites to produce climate-quality radiative fluxes at the top-of-atmosphere, within the atmosphere and at the surface, together with the associated cloud and aerosol properties.

The CERES project management and implementation responsibility is at NASA Langley. The CERES Science Team is responsible for the instrument design and the derivation and validation of the scientific algorithms used to produce the data products distributed to the atmospheric sciences community. The CERES DMT is responsible for the development and maintenance of the software that implements the science team's algorithms in the production environment to produce CERES data products. The Langley ASDC is responsible for the production environment, data ingest, and the processing, archival, and distribution of the CERES data products.

Document Overview

This document, CERES CATALYST Operator's Manual is part of the CERES CATALYST delivery package provided to the Langley Atmospheric Science Data Center (ASDC). It provides a description of the CERES CATALYST software and explains the procedures for executing the software. A description of acronyms and abbreviations is provided in Appendix A, a comprehensive list of messages that can be generated during the execution of the CATALYST server are contained in Appendix B, and the locations of the execution environment scripts are provided in Appendix C.

This document is organized as follows:

Introduction
Document Overview
CATALYST System Overview
Section 1.0 CATALYST Server
Section 2.0 Operator's Console

Appendix A - Acronyms and Abbreviations
Appendix B - Error Messages for CATALYST
Appendix C - CATALYST Environment File
Appendix D – List of Files Created on First Install
Appendix E – Product Availability in CATALYST Database

CATALYST System Overview

CATALYST is a semi-automated workflow manager designed and implemented by the CERES team to streamline production processing. It consists of a server and Operator's Console. The server is responsible for ingesting a PR sent from the PR Web Application, generating specific instances of the PGE within the PR, determining if all data inputs are available through the logging database, handling the submission and monitoring of PGE instances, and updating the logging database with status. The Operator's Console allows tracking of job instances and provides the ability for an operator to start and stop PGE instances. For user authentication to CATALYST and the PRDB, the *AMI* LDAP is used.

CATALYST Server

The CATALYST server is an application that provides a service to coordinate the execution of CERES PGEs in a production environment. The server runs as a multithreaded Perl daemon that provides a front-end XML-RPC API with which external applications (both Perl and non-Perl) can connect and query/modify the state of the CATALYST server. The CATALYST server ingests PRs sent from the CERES PR Tool. The system can accept all PRs for a given processing stream of interdependent PGEs. The system builds job instance information from ingested PRs and implements PGE specific logic to continuously poll for availability of input file preconditions through the logging database. CATALYST uses a logging database to store job instance information. It can also submit the epilogue scripts to archive output files.

A handler is needed for each PGE that CATALYST manages. The handler contains specific logic for creating job instances and determining its ability to run. A list of PGE handlers is included in Table 1-4.

CATALYST Operator's Console

The Operator's Console is a Java client side application that displays information from the CATALYST server and provides the ability to control individual PR execution and shut down of the server. The Operator's Console will show which PRs are active within CATALYST, PGE instances and their status, and *AMI* cluster node availability. An operator can pause, resume, delete, restart (or undelete) PGE instances. It also provides the ability to check PGE computer resource use. The operator can also shut down the server through the console. Each operator will run the console on their local workstations.

1.0 CATALYST Server

1.1 Server Details

1.1.1 Responsible Persons

Table 1-1. Subsystem Software Analysts Contacts

Item	Primary	Alternate
Contact Name	T. Nelson Hillyer	Joshua C. Wilkins
Organization	SSAI	SSAI
Address	1 Enterprise Parkway	1 Enterprise Parkway
City	Hampton	Hampton
State	VA 23666	VA 23666
Phone	757-951-1951	757-951-1618
Fax	757-951-1900	757-951-1900
LaRC email	thomas.n.hillyer@nasa.gov	joshua.c.wilkins@nasa.gov

1.1.2 E-mail Distribution List

E-mail distribution list can be obtained from the primary contact listed in Table 1-1.

1.2 Operating Environment

1.2.1 Runtime Parameters

Table 1-2. Runtime Parameters for Server

Parameter	Description	Data Type	Valid Values
-daemonize	Detach the server from the shell session to run continuously in the background.	NULL	

1.2.2 Environment Script Requirements

See Appendix C of this document for a detailed description of the CATALYST-specific environment.

1.2.3 Execution Frequency

Single Instance – This service runs in the background as a daemon; only a single instance of this program is intended to run in the production environment at any moment of time.

1.2.4 Memory/Disk Space/Time Requirements

Table 1-3. Memory/Disk Space/Time Requirements for CATALYST Server

System	System Name	Hardware	Total Run Time (HR:MN:SEC)	Memory (MB)	Disk Storage (MB)
	CATALYST	x86_64	Indefinite	Dependent on PR Load	Dependent on PR Load

1.2.5 PGE Handlers

Table 1-4. PGEs Handlers for the CATALYST Server

PGEName	Description
CER4.1-4.1P6	Imager Clear-Sky Determination and Clouds Detection, Cloud Pressure Retrieval, Cloud Optical Property Retrieval, and Convolution of Imager Cloud Properties with CERES Footprint Point Spread Function
CopyECS StartUp Macro	This is bundled with the PGE Handler for CER4.1-4.1P6. It produces the start up cloud masks required for starting CER4.1-4.1P6.
CER4.1-4.2P4	Edition4 Daily QC Processor
CER4.1-4.2P5	Edition4 Imager Clear Sky Map Update Processor
CER4.1-4.3P3	Edition4 Monthly QC Generator
CER4.5-6.1P4	Inversion to Instantaneous TOA Fluxes and Empirical Estimates of Surface Radiation Budget Subsystems 4.5 and 4.6 Main Processor (Terra)
CER4.5-6.1P5	Inversion to Instantaneous TOA Fluxes and Empirical Estimates of Surface Radiation Budget Subsystems 4.5 and 4.6 Main Processor (Aqua)
CER4.5-6.2P3	Subsystems 4.5 and 4.6 SSF Subset Postprocessor
CER4.5-6.4P2	Subsystems 4.5 and 4.6 Monthly Validation Site SSF Processor

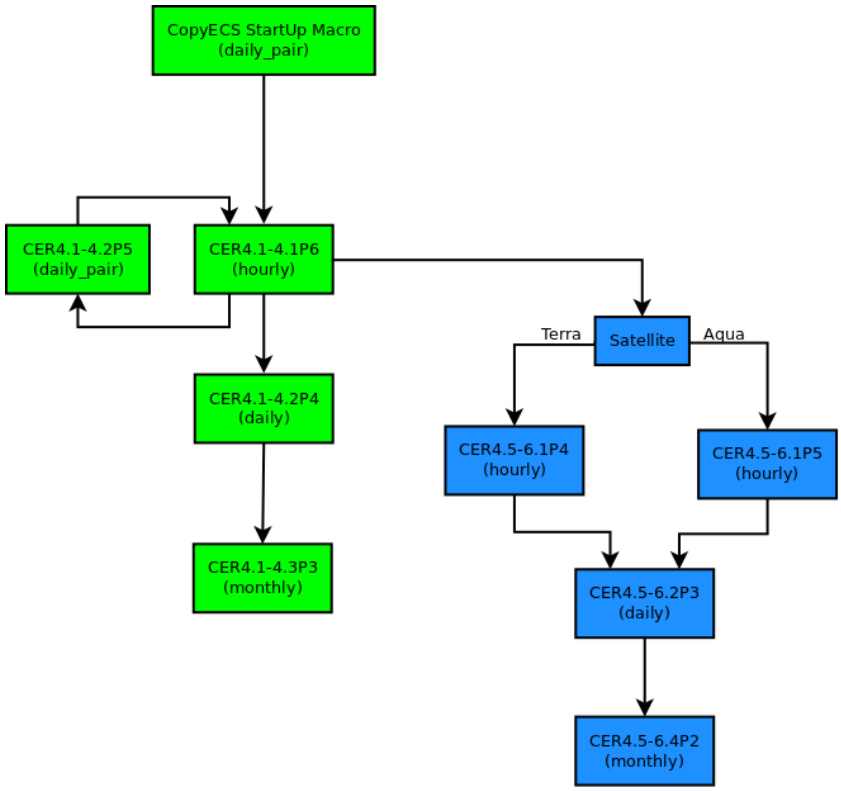


Figure 1-1. Flow Mapping for the CATALYST Server

1.3 Server Configuration Parameters

The server has several parameters that can be adjusted in "**$CATALYST_HOME/conf/catalyst.conf**". A summary of them, along with possible values are listed in the table below; ~~strikethrough text~~ denotes items highly suggested to remain there as delivered defaults (unless directed by the CATALYST development team):

Table 1-5. Configuration Parameters

Parameter	Purpose	Possible Values
~~cacher_server~~ *requires restart*	~~Not actively used since beta builds.~~	~~Leave in parameter file for 1.0.x releases.~~
catalyst_email_address *requires restart*	Suggested email address for sending CATALYST notifications.	Valid email address, e.g.: name@domain.com
~~connection_string~~ *requires restart*	~~PostgreSQL database connection parameters.~~	~~Leave alone. Only change under direction of CATALYST development team.~~
~~db_auth_file~~ *requires restart*	~~Path to PostgreSQL database connection authorization file containing database user name and password.~~	~~Path to properly formatted database auth file; the format is as follows:~~ ~~---begin file---~~ ~~Username~~ ~~Password~~ ~~---end file---~~

Table 1-5. Configuration Parameters

Parameter	Purpose	Possible Values
epilog_mode *no restart required if changed in Operator's Console*[1] *requires restart if manually edited*	The mode by which the epilogs are to run following PGE execution: "off" – neither manual or automatic "manual" – only when user submits command through Operator's Console "auto" – immediately after PGE execution completes	• "off" • "manual" • "auto"
epilog_path *requires restart*	Path to the ASDC epilog command to be used.	Valid path to ASDC epilog executable/script.
~~ldap_enabled~~ *requires restart*	~~Enable or disable the LDAP authentication routine. DISABLE WITH EXTREME CAUTION!~~	~~• "0" – disable~~ ~~• "1" – enable~~
~~ldap_server~~ *requires restart*	~~URL to the LDAP server (if ldap_enabled) to be used for client authentication.~~	~~Valid URL LDAP Server URL~~
logger_alarm_email_add *requires restart*	Email address(es) to send alarm messages about status of logging database connectivity.	Valid email address, e.g.: name@domain.com or, a space separated list of valid email addresses, e.g.: bob@email.com jim@email.com
~~mailx_implementation~~ *requires restart*	~~The implementation of the mailx command for the server running CATALYST.~~	~~• "bsd" - RHEL AMI-p~~ ~~• "heirloom" – SUSE AMI~~
max_job_count *requires restart*	The maximum number of "running" SGE jobs.	Integer > 0
pr_complete_email_add *requires restart*	Email address(es) to send notifications of PR completions or deletions.	Valid email address, e.g.: name@domain.com or, a space separated list of valid email addresses, e.g.: bob@email.com jim@email.com
server_name *requires restart*	Descriptive server name to be used for identification in Operator's Console displays.	Text
~~server_port~~ *requires restart*	~~TCP port to be used by the CATALYST server to accept client (PR Tool and Operator's Console) requests~~	~~Integer > 1024.~~
~~server_version~~ *requires restart*	~~Version number of CATALYST to be displayed in Operator's Console~~	~~Leave alone. Subject to removal in later releases.~~

[1] See 1.b.ii.3 on menu items in CATALYST Operator's Console.

1.4 Operating Procedures

1.4.1 Installing the CATALYST Server Software

This section describes how to install the CATALYST software. The installation procedures include an executable installation script which unpacks CATALYST software, configures run time environments, and adjusts the logging database as necessary.

1.4.1.1 Installation

1.4.1.1.1 Installation Instructions

Refer to Section 2.1 in the CATALYST Test Plan document for installation instructions.

Notes:

- Report any and all installation warnings/errors to the CATALYST development team (see Table 1-1) when encountered.

1.4.1.1.2 Installation Description

The CATALYST server is installed via a self-contained shell installer script. It performs the following basic functions: provides dialogs for choosing the desired environment and entering the installation key, extracting the server files, compiling RPC scripts, and setting up the logging database with the proper tables and base dataset. The files extracted by the installer script are listed in Appendix D.

Notes:

- The installer checks to see if a version of the CATALYST server is running before continuing with the install. It checks for the existence of **$CATALYST_HOME/bin/catalyst.pid** and aborts the install if found with a warning message to shut the server down before install.
- The base installer deletes the contents of the logging database (a warning dialog occurs during install in order to confirm database deletion) and removes any files in **$CATALYST_HOME/data/** . This prevents data clashes with previous installs of CATALYST.

1.4.1.1.3 Determining if the Installation is Successful

If all the files listed in Appendix D are present after running the installation and no errors were observed from the installation process itself, the installation was successful. The logging database can be checked to determine via SQL queries that it has the following tablenames (production_requests, pr_submissions, job_instances, and execution_hosts) and roughly greater than 8 million rows in the job_instances table.

1.4.1.1.4 Installing CATALYST Server Updates

Installing server updates is a very similar process to the base installer. It is also a self-contained installer which updates only the server code and related scripts, but unlike the base installer it is much more lightweight and does not require the user to choose an install environment. Furthermore, it does not interact with the logging database and does not modify the contents of $CATALYST_HOME/data/ .

1.4.1.1.5 Installing Logging Database Patch Scripts

Patch scripts will be delivered to add dependency data to the logging database on occasion. These will be self-contained scripts similar to the installer and will inject data into the database when run. The naming convention for the script is as follows: catalyst_db_patch_vX.X.X-XXXX where vX.X.X is the CATALYST version number the patch is associated with and XXXX is the SCCR number associated with the delivery. Success of a patch is determined if no errors were encountered when running the patch script. An additional spot check can be done using the provided set of SQL queries to verify several expected jobs (contained in the catalyst_db_patch_vX.X.X-XXXX.sql file delivered with the patch installer script) are contained in the database after the patch has run.

To install the database patch take the following steps (note – the patch install script must be run from the **$CATALYST_HOME** directory):

```
> source $CERESENV
> cd $CERESHOME/catalyst
> setenv CATALYST_HOME $PWD
> ./catalyst_db_patch_vX.X.X-XXXX.sh
```

Wait for the script to run and then verify there were no errors.

To use the sql spot checks provided, run the following command filling in the bracketed parameters with the proper values for the environment where **$CATALYST_HOME** is located (e.g. PPE):

```
> psql --set ON_ERROR_STOP=on -h [database server] -p [database port] -U [database
  user] [database name] < $CATALYST_HOME/catalyst_db_patch_vX.X.X-
  XXXX.sql
```

The queries ran successfully if each query returns a value and no errors were produced.

1.4.2 How to Start the CATALYST Server

The CATALYST server can be started three different ways. The rationale for the different execution methods are in the following subsections. In all cases, the STDOUT and STDERR output for all three methods is sent to the following file:
$CATALYST_HOME/log/catalyst.log

1.4.2.1 Interactive

The CATALYST server will run in the foreground in a user's shell either inside or outside of a terminal multiplexer session (like GNU Screen). This is the method that has been employed in CM and SIT testing thus far. In a testing sense, this is convenient to be able to see server-side console output (errors, warnings, etc.) and quickly kill the server. In a production sense, this puts a high reliance on the shell session not being terminated by human error or by network connection dropouts. **This is not recommended for a production setting!**

The server can be started interactively by following these steps:

> **source $CERESENV**
> cd $CERESHOME/catalyst
> setenv CATALYST_HOME $PWD
> cd bin
> ./run_catalyst_server.sh

The server will run indefinitely in the shell session until the server has been shutdown or terminated.

1.4.2.2 Detached/Daemonized

The CATALYST server will run as a process that is detached from the user's shell session (also referred to as a daemonized process). This mode of execution eliminates the reliance on a continuously maintained shell session like in Section 1.4.2.1. However, this method does not provide a direct means to automatically restart the CATALYST server after a system reboot.

The server can be started in a daemonized mode by following these steps:

> **source $CERESENV**
> cd $CERESHOME/catalyst
> setenv CATALYST_HOME $PWD
> cd bin
> ./run_catalyst_server.sh -daemonize

The server is now detached from the shell session. You may now log out of your terminal window without killing the CATALYST server.

1.4.2.3 Linux System Service (init.d entry)

This is very much like Section 1.4.2.2 with several added benefits:

- The CATALYST service can automatically start on system boot.
- The CATALYST service can automatically shutdown on system halt.
- The CATALYST service can be administered like other COTS software by regular system administrators. Commands like the following will be made available for administrative users for starting, stopping, restarting, and querying status, respectively:
    - service catalyst start
    - service catalyst stop
    - service catalyst restart
    - service catalyst status

If this method is preferred for production, an auxiliary delivery with the necessary software and installation directions will be provided upon request.

1.4.3 How to Verify the CATALYST Server is Running

The following set of commands will verify that the CATALYST server is running. The steps involve checking your *AMI* username and password (prompted for entry in the following steps) against the *AMI* LDAP server. These steps replicate the process the server implements when running in the production environment as the same XML-RPC API calls are utilized.

At the command-line (>) type:

> source $CERESENV
> cd $CERESHOME/catalyst/conf
> source catalyst_env.csh
> cd $CERESHOME/catalyst/test_suites
> ./test_login.pl

The "**test_login.pl**" application will prompt the user for your *AMI* username and password. Upon entering the username and password, the system will indicate that the following actions have been performed:

1. Logged into server and received a token.
2. Verified the server acknowledges the token as being valid.
3. Logged out of the server.
4. Verified that the server no longer acknowledges the token as being valid.
5. Verified that a falsified token is not accepted by the server.

If steps 1-5 have completed successfully, then the server is functioning as intended.

1.4.4 Editing the CATALYST Server's Access Control List

To restrict the abilities of various users, CATALYST utilizes an Access Control List (ACL). This ACL describes what privileges a particular user has in CATALYST: admin, manager, operator, or a combination of the three. Table 1-6 describes the rights granted by the three security categories:

Table 1-6. CATALYST Privilege Categories

Name	Abilities
admin	Gives the user the ability to shutdown the server.
manager	Gives the user the ability to delete PRs and mark PRs complete.
operator	Gives the user the ability to submit PRs and pause, resume, delete, force start, and resubmit jobs and/or epilogs.

The ACL file is located in "**$CERESHOME/catalyst/conf/users.conf**". The file is a plain text file with one user (represented by their *AMI* username) per line. Following the user's name is a list of the privilege categories to which they belong. For example:
thillye1 operator

grants thillye1 only operator privileges, while:

jcwilki1 operator manager

grants jcwilki1 both manager and operator privileges.

NOTE: Users not listed in the CATALYST ACL are given read-only privileges to the CATALYST server; they will not be able to modify the state of the server or jobs managed by the server. Only users with AMI-p login privileges will be able to interact with the CATALYST server.

1.4.5 Terminating the CATALYST Server

In the event the CATALYST server must be shutdown, a CATALYST administrator can initiate the shutdown command via CATALYST Server -> shutdown in the Operator's Console (it may take several moments to shutdown). *In the event this does not work, take the following steps*:

At the command-line (>) type:

```
> cd $CERESHOME/catalyst/bin
> cat catalyst.pid | xargs kill && rm catalyst.pid
```

The server is now terminated.

1.5 Processing Request Submission from the Command Line

In the event the Processing Request (PR) Web Application is unavailable and PRs must be submitted, an alternative submission method exists. Since CATALYST is API-based, submitting PRs from the command line performs and behaves the same as if submitted via the PR Web Application.

1.5.1 Creating a PR Load Script

This functionality is intended to be used for testing only! While CATALYST accepts PRs from these utility scripts or from the PR Tool, the PR Tool is the definitive and approved source of PRs and should be used to submit PRs to CATALYST in the production environment.

See the '**test_load_CERX_XPX.pl**' scripts in the **$CATALYST_HOME/test_suites** directory for an example on how this is done. The file consists of the following primary sections:

- **identifier:** This section of the file is where the PR identification information is placed. All of these values must exist in order to associate this submission with the PR Web Application.
- **instances:** This section describes the datadate range for this PR submission.
- **environment:** This section provides the environment variables that apply for the defined instances. Environment variables that are different for a subset of the datadates must exist in their own PR file.
- **arguments:** This section provides any optional arguments that may be required for running this PGE. Oftentimes, this field is empty.

1.6 Expected PR Instantiation Time

The amount of time required for CATALYST to generate jobs from a PR is dependent on the following factors:

- The CPU and I/O load of the host and/or disk storage system (this is especially true in an environment where resources are shared amongst CATALYST and non-CATALYST operations)
- The size of the PR that is to be loaded.
- The amount of other PRs which have been or are being loaded.

Based on the above factors, the exact amount of time for loading a PR into CATALYST can vary as its runtime is a non-deterministic process. To alleviate some of this, CATALYST does perform some task switching between PRs as it create jobs and perform predecessor background checks so as to maximize execution progress across the scope of active PRs. Note that it is **not** required to load a PR in its entirety before being able to process the workunits comprising that PR.

In an ideal environment with no system load and no other PR activity, one will find that within ~30 seconds, there will be workunits marked as **waiting** and/or **scheduled/executing**.

1.7 Execution Evaluation

1.7.1 Exit Codes

The CATALYST server can exit with the codes listed in Table 1-7. Exit codes marked as failure should be reported to the responsible persons listed in Table 1-1.

Table 1-7. Exit Codes for CATALYST Server

Exit Code	Definition	Action
0	Normal Exit	Server started normally
Other	Failure	Notify persons listed in Table 1-1.

Exit codes from PGEs run inside of CATALYST should be compared against the CERES *Standard Exit Codes* document found on the CERES documentation website.

1.8 Operational Log Files/Directories.

CATALYST updates several log files during operation.

Table 1-8. Log File Listing for CATALYST Server

Directory	File Name	Purpose
$CERESHOME/catalyst/log	main.log	Lists activity relating to server startup, runtime (detected warnings/errors), and shutdown.
$CERESHOME/catalyst/log	transactions.log	Lists requests made by users that modify the state of the server or jobs maintained by the server.
$CERESHOME/catalyst/log	database.log	Lists errors/failures in connections to external database(s).
$CERESHOME/catalyst/log	email.log	Lists outgoing email subjects.

Table 1-8. Log File Listing for CATALYST Server

Directory	File Name	Purpose
$CERESHOME/catalyst/log	catalyst.log	Lists diagnostics/warnings/errors originating from the Perl interpreter. Also, currently lists macro activity (subject to change in later builds).
$CERESHOME/catalyst/log/epilogue/$PGE/$YYYY/…	$CATALYST_JOB_NAME.epilogue	Contains the STDOUT/STDERR from the ASDC epilogue process that is called by CATALYST.

1.9 Operational Files/Databases.

The table below outlines files that CATALYST produces. These files should **never** be removed and are part of the normal operation of CATALYST.

Table 1-9. Operational File Listing for CATALYST Server

Location	Name
$CERESHOME/catalyst/data	all files with the extension .sqlite3
$CERESHOME/catalyst/data	all files with the extension .sqlite3-shm
$CERESHOME/catalyst/data	all files with the extension .sqlite3-wal
$CERESHOME/catalyst/data	all files with the extension .socket
dsrvr205.cluster.net	This server hosts the CATALYST Logging Database. Later builds of CATALYST will provide a database inspection API. ceres_logdb_ppe *(only when running in PPE)* ceres_logdb_prod *(only when running in Production)*

The CATALYST databases on "dsrvr205.cluster.net" are initially populated with static data included in the CATALYST installer. They contain records of the existence and approved non-existence of predecessor products used by the PGE Handlers in Table 1-4. Future versions of CATALYST will allow dynamic updates of predecessor products in the CATALYST databases.

1.10 CATALYST Job Life Cycle

Jobs in CATALYST have states which describe the current phase of execution that job is performing. There are also events that are automatic (i.e., default behavior) and manual events (i.e., from the operator via the Operator's Console) that transition CATALYST jobs from one state to the next.

1.10.1 Job States

- **Unitialized** – This is the starting state for a job. CATALYST has generated this job and its predecessor list using information contained in the PR. This job now needs to obtain a CATALYST job ID.

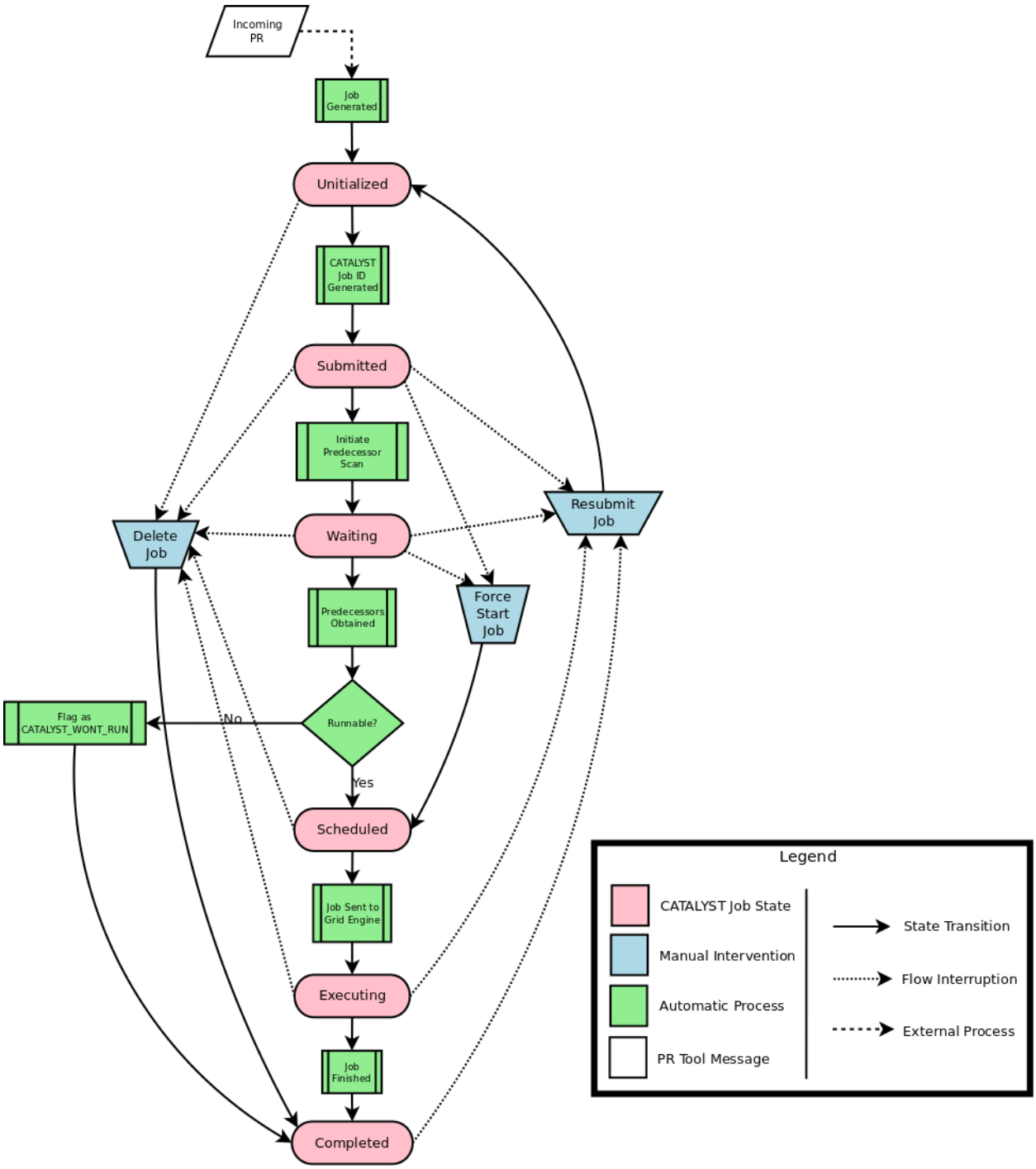
- **Submitted** – This job has been registered with the CATALYST logging database and has been assigned a CATALYST job ID.
- **Waiting** – This job is now performing two core functions: scanning the logging database for its predecessors and listening for its predecessors to complete. The events happening behind the scene in this state are what prevent a job from executing prematurely. One can expect a job to be in this state for the most amount of its *non-completed* lifespan.
- **Scheduled** – The job has completed all of its predecessor checks and is now ready to be scheduled to the grid engine. Any job in this state can be safely marked for execution at any time.
- **Executing** – The CATALYST job is now runnable in the cluster's grid engine. The grid engine makes the decision to wait or run based on available compute resources and other non-CATALYST activity. The job will remain in this state until it has completed running and CATALYST has recorded its grid engine exit status.
- **Completed** – This is the terminating state for jobs under CATALYST.

1.10.2 Manual Actions Influencing State Transitions

- **Delete Job** – The delete job action coerces the job to enter the **completed** state. This action can be performed at all states with the exception of the **completed** state.
- **Resubmit Job –** The resubmit job action transitions a job back to the starting state for CATALYST: **uninitialized**. This action can be performed at all states with the exception of the **uninitialized** state.
- **Force Start Job –** The force start job action transitions a job to the **scheduled** state. This action can be performed at the following states: **submitted** and **waiting**.

1.10.3 Graphical Workflow

This is a graphical representation of the states and state transitions. Pause and resume are intentionally left out since they don't influence state changes, they instead lock the job into its current state. See the section on "Automatic Determination of Will Not Run Conditions" for a discussion of CATALYST_WONT_RUN.



1.11 Automatic Determination of Will Not Run Conditions

There are several cases where CATALYST can make an automatic determination where sufficient amounts of input products are not available. These input products could be products external to CATALYST such as MODIS granules, or internal to CATALYST such as CRHU files produced by CER4.1-4.1P6. The will not run conditions are signaled by CATALYST with the exit status of CATALYST_WONT_RUN (described in the CERES Standard Exit Codes document). The missing external products must be labeled in the CATALYST logging database

as CATALYST_NOT_PLANNED status in order to signify that this product is missing for an acceptable missing, not just unaccounted for without reason.

Below are the cases where CATALYST will evaluate CATALYST_WONT_RUN conditions (refer to the flowchart in Table 1-4 for PGE granularity):

- **CER4.1-4.1P6:**
  - No matching pairs of 5 minute MOD02 (or MYD02)[2] and MOD03 (or MYD03) granules found for this data hour[3].
- **CER4.1-4.2P4:**
  - No output from the 24 hourly predecessor CER4.1-4.1P6 jobs was created for this data date.
- **CER4.1-4.3P3:**
  - No output from the CER4.1-4.2P4 jobs was created for this data month.
- **CER4.5-6.1P4/5[4]:**
  - No output from the CER4.1-4.1P6 hour matching this data hour.
  - CER4.1-4.1P6 was run without one or both IES(s) matching this data hour.
- **CER4.5-6.2P3:**
  - No output from the CER4.5-6.1P4/5 hourly jobs was created for this data dates.
- **CER4.5-6.4P2:**
  - No output from the CER4.5-6.2P3 data dates for this data month.

[2] For MODIS granules, the product naming convention for Terra MODIS is MODxx and Aqua MODIS is MYDxx (where xx is the MODIS product number).
[3] MOD04 (or MYD04, see footnote above) granules are treated as optional for this PGE. The only requirement that CATALYST has for these files is that a record of their existence (i.e. SUCCESS) or non-existence (CATALYST_NOT_PLANNED) be recorded. The job will not start until such a record in the logging database exists; release 1.1 and later will provide features to do this, however in the meantime, CATALYST has been delivered with the MOD04 granules recorded sufficiently for Edition4-Beta Clouds and Inversion processing.
[4] In case Inversion is not run directly after Clouds, the CER4.5-6.1P4/5 CATALYST_WONT_RUN determination will not work, and will result in JSS_ERRORs. These JSS_ERRORs jobs must be deleted, since they will not run regardless of execution under CATALYST.

1.12 Best Practices

There are several best practices for efficiently using the CATALYST software for CERES data production:

- **Submit PR chunks as soon as possible to ensure CATALYST can keep busy during staff downtime**. Not keeping CATALYST fed with enough workunits will cause starvation which has negative impacts on processing speed. This also allows CATALYST the opportunity to interleave a PR's PGE execution and background checks for the PR itself. The background checks specifically for CATALYST operation can take several minutes to complete, so performing this step as early as possible lessens the impact of this.
- **Avoid "block running" PR chunks.** For example, submitting and running an entire month of one PR before submitting a month of a successor PR. This may be convenient for executing PRs by hand, but it is a very inefficient strategy in the long-run because it results in large idle periods in both the predecessor and successor PRs. CATALYST knows how to run jobs in a PR in the CERES subsystem approved order. For example:
    - Run the CER4.1-4.1P6, CER4.1-4.2P5, CER4.1-4.2P4, CER4.1-4.3P3, CER4.5-6.1P4, CER4.5-6.1P5, CER4.5-6.2P3, CER4.5-6.4P2 PR stream concurrently. This helps to maximize cluster utilization, and minimizes CATALYST start up costs.

1.13 Caveats

This section describes known bugs or issues with the CATALYST software and provides a means to work around them, if possible. Verify you are using the **most recent version of this document** since these caveats are subject to change with findings over time or with new CATALYST software updates.

1.13.1 PGE Execution

1.13.1.1 Periodic JSS_ERRORs during Delayed Execution of Inversion CER4.5-6.1P4/5

There is a known issue when there is a delay between the execution of Inversion CER4.5-6.1P4 and CER4.5-6.1P5 jobs and their predecessor CER4.1-4.1P6 jobs in the event the CER4.1-4.1P6 job was run with one or both IES inputs missing.

When an IES is not available for an hour of CER4.1-4.1P6, the intermediate SSF for that instrument cannot be produced. When this occurs, CER4.1-4.1P6 exits with a status of SUCCESS_WQ. SUCCESS_WQ is thrown for one IES missing or both IESs missing. From the perspective of the CER4.5-6.1P4/5 job looking at CER4.1-4.1P6's SUCCESS_WQ status, it is ambiguous to CATALYST if one or both instruments' IESs were missing.

CATALYST at this point, must look beyond solely the exit code of CER4.1-4.1P6 to intelligently determine if one or both instrument runs of CER4.5-6.1P4/5 can successfully run (i.e. did CER4.1-4.1P6 produce one set or neither set of intermediate SSFs). CATALYST does this by having the inversion job introspect the clouds job, where the IES existence information is held. The clouds job that is being inspected is done using the full set of on-disk data at the time the clouds job completes. This workflow is fine when inversion and clouds are run back-to-

back; in the sense of the inversion job is in the **waiting** state when the clouds job transitions to the **completed** state. With this IES information, inversion can then either run, or flag itself with CATALYST_WONT_RUN (see the section titled "Automatic Determination of Will Not Run Conditions).

This workflow, unfortunately, breaks down when inversion is set to run long after the clouds job has run. At that point, when the inversion job investigates the clouds job's status, it does so using the logging database subset of CATALYST job data. The logging database subset of CATALYST job data does not contain the individual job's product tables used for determining the individual job's execution firing order. Without this data, the inversion job cannot make the distinction as to what the SUCCESS_WQ from clouds actually means (is it due to one or both IESs missing), so at this point the only thing to do is to try to run the job and suffer the consequences due to lack of detailed information.

At this point, if the clouds job did not produce the intermediate SSF, the inversion job will fail to start, thereby giving a JSS_ERROR. The only thing to do at this point is to do the following:

1. Identify the datadate and instrument of the inversion job (CER4.5-6.1P4/5) that failed.
2. Locate the predecessor clouds job (CER4.1-4.1P6) of that inversion job and examine the IESs present in the job details window (in the Operator's Console).
3. If the IES is not flagged as SUCCESS for that instrument, then you'll have to resubmit and immediately delete the inversion job. There's nothing of value lost from the inversion job, since it will not run even if it was not running in CATALYST. Why resubmit first? CATALYST will not let you delete a completed job, so you have to delete the job in the pre-**completed** stages.
4. Deleting the hourly inversion job (CER4.5-6.1P4/5) will then release the daily CER4.5-6.2P3 job for execution, if it was stuck waiting for only that job to signal completion.
5. In the event the CER4.5-6.2P3 (and/or CER4.5-6.4P2) jobs are not released when it should, that job can be resubmitted using the CATALYST Operator's Console.

The likelihood of this issue occurring can be significantly reduced (if not prevented) by following the CATALYST "Best Practices" section and also ensuring that any and all ancillary data is in the CERES subsystem's workspace, which go to compound this issue with other problems.

The next (non-patch) release of CATALYST will include features that should effectively eliminate this problem from occurring.

1.13.2 Loss of Database Connectivity

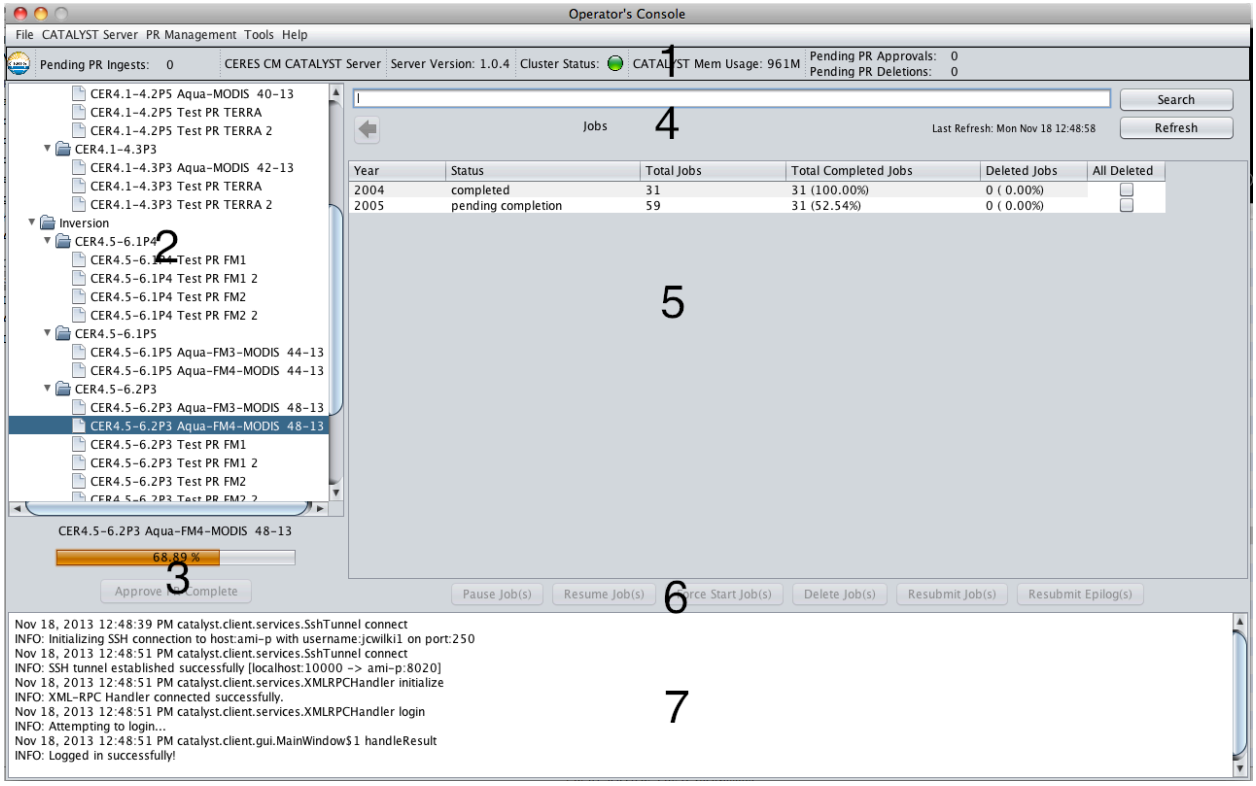
In the event that the CATALYST server loses connectivity to the database server and does not recover, the server can be restarted to restore the database connectivity. Some ways to determine if this has happened: the main server log (catalyst.log) should contain errors about the database if it could not restore connection, and job transitions would seem to be stuck (not going from waiting → scheduled or from submitted → waiting, or the like) and not moving to the expected state.

2.0 Operator's Console

Table 2-1. Dependencies

Dependency	Version Supported
JAVA Runtime	Version 6 or Newer

2.1 Layout and Features



1. Menu Bar – Contains various menu options and a toolbar containing server status information:
    a. Toolbar Statuses (left to right)
        i. Pending PR Ingests - A counter that displays how many PRs are currently being ingested by the server.
        ii. Server Name – Shows the name of the server designated in the server configuration
        iii. Server Version – The build version number of the server
        iv. Cluster Status – Shows whether the SGE cluster is ready to accept new jobs

v. CATALYST Memory Usage - Shows the current memory footprint of the server

vi. Pending PR Approvals: - Counter for how many PRs are currently in the process of being approved and removed from the server

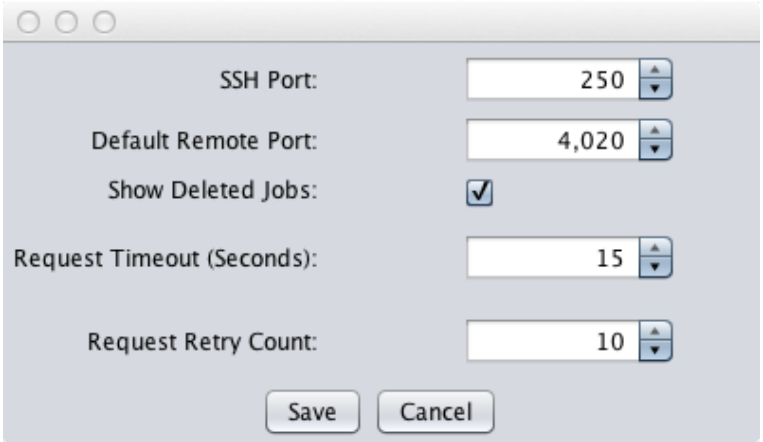
vii. Pending PR Deletions – Counter for how many PRs are currently in the process of being deleted and removed from the server

b. Menus

i. File

1. Preferences – Displays a dialog to specify the SSH Port, Default Remote Port, if deleted jobs are to be shown in the Operator's Console, the timeout in seconds for requests sent to the server, and the maximum number of retry attempts for outstanding requests to the server before the Operator's Console gives up.

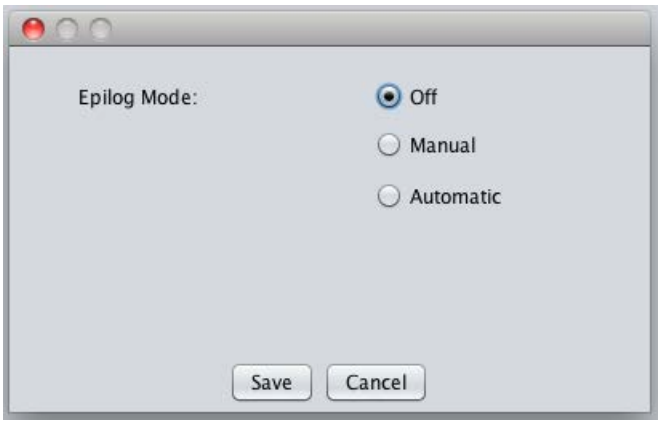
   Ideally the timeout and retry count preferences do not need to be changed. If disconnetions from the Operator's Console to the CATALYST server become a frequent occurance, contact the CATALYST development team for guidance on how to proceed. The preferences are in place so that they can be changed from the developer-recommended settings without the Operator's Console needing to be redelivered.



2. Exit – Closes the Operator's Console. DOES NOT SHUTDOWN THE CATALYST SERVER.

ii. CATALYST Server

1. Connect – Establish a connection to the CATALYST Server.
2. Disconnect – Disconnects from the CATALYST Server.
3. Epilog Mode – Opens up a dialog that lets a CATALYST manager adjust the overall epilog mode of the server (off, manual, auto)



4. Reload ACL – Reload the CATALYST Server's Access Control List.
5. Shutdown – Shutdown the CATALYST Server. This operation can only be performed by a CATALYST Administrator.

iii. PR Management

1. Delete Selected PR from CATALYST – Delete an actively running PR from CATALYST. This operation can only be performed by a CATALYST Manager.

iv. Tools

1. Blade List – Opens up a dialog window containing the list of blades. This list takes a few seconds to populate the first time it is opened.

v. Help

1. Available RPC Functions – Displays a list of the XML-RPC functions exported by the CATALYST server.
2. Available PGE Handlers – Displays a list with details about the current active PGE Handlers in the CATALYST server.
3. Server Properties – Shows the current server uptime and the time the server was started
4. About – Displays an about dialog with some contact information.

2. Processing Request List – Tree of processing requests currently running in CATALYST, organized by subsystem → PGE → Production Request. Double click a PR to retrieve more details in a separate window. Right clicking a PR will show a delete PR option, this function is only available to CATALYST managers.
3. PR Progress Bar – Shows the percentage complete of the currently selected PR. Once 100% completion is reached, the Approve PR Complete button will become available. This allows the selected PR to be cleared from CATALYST as having been completed. When selecting a different PR the progress bar may show an indeterminate animation until it retrieves information back from the server.

4. Job Navigation – Provides navigational buttons for refreshing the currently displayed jobs, or moving back up to the root with the "back" button. The search feature allows for listing jobs from the selected PR that match the given datadate. Input a datadate to do a search with this format: yyyymmddhh (broad searches can be made by only inputting a partial datadate – e.g., 201002 will display the days in February 2010).
5. Job Table – Lists the jobs from the selected PR and their current status in CATALYST. Select a job by left clicking. Once selected, a right-click menu is available for the selected jobs. Navigate deeper into the jobs tree by double clicking a date. During times of high load on the CATALYST server host system, the Job Table will show a single row with the text "Fetching Data" to show it is waiting for data to come back from the server. Double clicking a job at the lowest level brings up the details for that job in a window.
6. Job Actions – Use these buttons to perform actions on the selected job(s).
    a. Actions can be performed on single jobs or multiple highlighted jobs.
    b. Actions can also be performed on the job folder level. This method of action applies as follows for different PGEs:
        i. Hourly – Folder actions can apply for the day.
        ii. Daily – Folder actions can apply for the month.
        iii. Daily_Pair – *not supported*.
        iv. Monthly – *not supported*.
        v. Monthly_Zonal – *not supported*.
7. Log Window – Any errors, warnings, or info from the Operator's Console are displayed in this scrolling window, as well as put into a log file.

2.2 Obtaining the Operator's Console Software

The latest version of the Operator's Console can be downloaded from the CATALYST Home page at the following URL:

http://ceres.larc.nasa.gov/Internal/intern_docs_catalyst.php

Download either the zip or tar file of the latest Operator's Console (choose the format you are most comfortable with) and extract it. Inside will be a README file containing further instructions and information. A summary of that information is outlined below:

The Operator's Console requires at least the JAVA 6 (or newer) runtime installed and kept up to date.

To run the Operator's Console choose from the following based on your operating system:

1. Windows → double click start_oc.bat or double click OperatorsConsole.jar
2. Mac → double click OperatorsConsole.jar or run ./start_oc.sh from the command line
3. Linux → run ./start_oc.sh from the command line

Once the console starts up, you are prompted to enter in the server details for CATALYST. The server will typically be *AMI-P*, but if you are not sure ask. The username and password will be the one you use to login to the *AMI* systems. The Operator's Console generates a log file as well as displays that log at the bottom of the console window. If you notice any unexpected errors, please send the log file and a description of what you were doing to the CATALYST development team.

2.3 Recovering from a Operator's Console Failure

If the Operator's Console becomes unresponsive or has a failure that results in it being inoperable, there are several steps to take:

1. Click "CATALYST Server → Disconnect" followed by trying to reconnect to the server "CATALYST Server → Connect" and re-enter the corresponding login information.
2. Close and restart the Operator's Console. This should clear up any unresponsiveness.
3. Check that the login and server information is correct and that the server is running.
4. If any error messages appear in the console at the bottom or restarting the client does not work to solve your issue, please email the most recent client.log file generated by the console to joshua.c.wilkins@nasa.gov or your designated contact for CATALYST related bugs with a detailed description of what happened.

Appendix A
Acronyms and Abbreviations

ACL	Access Control List
API	Application Programming Interface
ASDC	Atmospheric Science Data Center
CATALYST	CERES AuTomAted job Loading sYSTem
CERES	Clouds and the Earth's Radiant Energy System
CM	Configuration Management
COTS	Commercial Off The Shelf
LaRC	Langley Research Center
LDAP	Lightweight Directory Access Protocol
NASA	National Aeronautics and Space Administration
PR	Processing Request
SSAI	Science Systems and Applications, Inc.
XML-RPC	Extensible Markup Language – Remote Procedure Call

Appendix B
Error Messages for CATALYST

Error/Warning Message	Diagnosis
Can't locate object method "timeout" via package "RPC::XML::Server::new: Unable to create HTTP::Daemon object: HTTP::Daemon: Address already in use" (perhaps you forgot to load "RPC::XML::Server::new: Unable to create HTTP::Daemon object: HTTP::Daemon: Address already in use"?) at ./catalyst_server.pl line 61.	This message implies that there is already an instance of the CATALYST server running on your host. If the other CATALYST server that is running is no longer needed, follow the CATALYST server termination steps in Section 1.4.5.

Appendix C
CATALYST Environment

The execution environment required to run CATALYST on the target system can be configured by executing all of the following commands:

> source $CERESENV
> setenv CATALYST_HOME $CERESHOME/catalyst
> cd $CATALYST_HOME/conf
> source catalyst_env.conf

Appendix D
List of Files Created On First Install

bin/
bin/catalyst_server.pl
bin/drmaa.h
bin/run_catalyst.sh
conf/
conf/catalyst_env.csh
conf/cm_catalyst.conf
conf/handlers.xml
conf/handlers.xsl
conf/mapfile.conf
conf/ppe_catalyst.conf
conf/production_catalyst.conf
conf/rpc_permissions.conf
conf/users.conf
conf/auth_cm.gpg
conf/auth_ppe.gpg
conf/auth_prod.gpg
temp/
monitors/
monitors/tattletale_offline_submit.pl
monitors/tattletale_v1.pl
data/
scripts/
scripts/drmaa_controller.pl
scripts/drmaa_interface.pl
scripts/dummy_epilog.pl
scripts/epilog_submit_wrapper.pl
scripts/grab_blade_list.pl
scripts/grab_ganglia.pl
scripts/grab_uge_hostname.pl
scripts/process_facade.sh
scripts/tattletale_submit.pl
handlers/
handlers/CER4_1_4_1P6_PGE_Handler.pm
handlers/CER4_1_4_2P4_PGE_Handler.pm
handlers/CER4_1_4_2P5_PGE_Handler.pm
handlers/CER4_1_4_3P3_PGE_Handler.pm
handlers/CER4_5_6_1P4_PGE_Handler.pm
handlers/CER4_5_6_1P5_PGE_Handler.pm
handlers/CER4_5_6_2P3_PGE_Handler.pm
handlers/CER4_5_6_4P2_PGE_Handler.pm
lib/DataDate/
lib/DataDate/Daily.pm
lib/DataDate/DailyPair.pm
lib/DataDate/DataDate.pm
lib/DataDate/Hourly.pm
lib/DataDate/Monthly.pm
lib/DataDate/MonthlyZonal.pm
lib/libexpat.so.1
lib/libexpat.so
lib/Cacher.pm
lib/CERES_TLS.pm
lib/ClusterResourceMonitor.pm
lib/ColdStorage.pm
lib/DBFILE_INIT.pm
lib/DRMAA.pm
lib/EmailManager.pm
lib/Handler.pm
lib/HotStorage.pm
lib/JobContainer.pm
lib/Job.pm
lib/LoggingDbHandler.pm
lib/LoggingDbWriter.pm
lib/LogUtils.pm
lib/MainHandler.pm
lib/PGE_Job.pm
lib/POSIX_SYSCALL.pm
lib/PreferencesHandler.pm
lib/ProductionRequest.pm
lib/ProductPGEMapper.pm
lib/Product.pm
lib/ResourceParser.pm
lib/ReturnCodes.pm
lib/RPCUtils.pm
lib/Scheduler.pm
lib/StorageLocker.pm
lib/TermUtils.pm
lib/ThreadManager.pm
lib/TKUI.pm
lib/User.pm
lib/Makefile
lib/xpl/
lib/xpl/Makefile
lib/xpl/cps/
lib/xpl/cps/Makefile
lib/xpl/cps/getHandlers.base
lib/xpl/cps/getHandlers.code
lib/xpl/cps/getHandlers.help
lib/xpl/cps/getMemoryUsage.base
lib/xpl/cps/getMemoryUsage.code
lib/xpl/cps/getMemoryUsage.help
lib/xpl/cps/getPreferences.base
lib/xpl/cps/getPreferences.code
lib/xpl/cps/getPreferences.help
lib/xpl/cps/getProperties.base
lib/xpl/cps/getProperties.code
lib/xpl/cps/getProperties.help
lib/xpl/cps/reloadACL.base
lib/xpl/cps/reloadACL.code
lib/xpl/cps/reloadACL.help
lib/xpl/cps/setEpilogMode.base
lib/xpl/cps/setEpilogMode.code
lib/xpl/cps/setEpilogMode.help
lib/xpl/cps/setMaxJobCount.base
lib/xpl/cps/setMaxJobCount.code
lib/xpl/cps/setMaxJobCount.help
lib/xpl/cps/shutdown.base
lib/xpl/cps/shutdown.code
lib/xpl/cps/shutdown.help
lib/xpl/cps/shutdownCalled.base
lib/xpl/cps/shutdownCalled.code
lib/xpl/cps/shutdownCalled.help
lib/xpl/crm/
lib/xpl/crm/Makefile
lib/xpl/crm/getBladeResources.base
lib/xpl/crm/getBladeResources.code
lib/xpl/crm/getBladeResources.help
lib/xpl/crm/getBlades.base
lib/xpl/crm/getBlades.code
lib/xpl/crm/getBlades.help

lib/xpl/crm/getClusterResources.base
lib/xpl/crm/getClusterResources.code
lib/xpl/crm/getClusterResources.help
lib/xpl/crm/isBladeReady.base
lib/xpl/crm/isBladeReady.code
lib/xpl/crm/isBladeReady.help
lib/xpl/crm/isClusterReady.base
lib/xpl/crm/isClusterReady.code
lib/xpl/crm/isClusterReady.help
lib/xpl/pge/
lib/xpl/pge/Makefile
lib/xpl/pge/completePR.base
lib/xpl/pge/completePR.code
lib/xpl/pge/completePR.help
lib/xpl/pge/createJobs.base
lib/xpl/pge/createJobs.code
lib/xpl/pge/createJobs.help
lib/xpl/pge/deleteInstances.base
lib/xpl/pge/deleteInstances.code
lib/xpl/pge/deleteInstances.help
lib/xpl/pge/deletePR.base
lib/xpl/pge/deletePR.code
lib/xpl/pge/deletePR.help
lib/xpl/pge/executeMacro.base
lib/xpl/pge/executeMacro.code
lib/xpl/pge/executeMacro.help
lib/xpl/pge/forceRestartJobs.base
lib/xpl/pge/forceRestartJobs.code
lib/xpl/pge/forceRestartJobs.help
lib/xpl/pge/getHandlerDetails.base
lib/xpl/pge/getHandlerDetails.code
lib/xpl/pge/getHandlerDetails.help
lib/xpl/pge/getHandlers.base
lib/xpl/pge/getHandlers.code
lib/xpl/pge/getHandlers.help
lib/xpl/pge/getPendingCompletes.base
lib/xpl/pge/getPendingCompletes.code
lib/xpl/pge/getPendingCompletes.help
lib/xpl/pge/getPendingDeletes.base
lib/xpl/pge/getPendingDeletes.code
lib/xpl/pge/getPendingDeletes.help
lib/xpl/pge/getPendingIngests.base
lib/xpl/pge/getPendingIngests.code
lib/xpl/pge/getPendingIngests.help
lib/xpl/pge/pauseInstances.base
lib/xpl/pge/pauseInstances.code
lib/xpl/pge/pauseInstances.help
lib/xpl/pge/queryJobDetails.base
lib/xpl/pge/queryJobDetails.code
lib/xpl/pge/queryJobDetails.help
lib/xpl/pge/queryJobs.base
lib/xpl/pge/queryJobs.code
lib/xpl/pge/queryJobs.help
lib/xpl/pge/queryPR.base
lib/xpl/pge/queryPR.code
lib/xpl/pge/queryPR.help
lib/xpl/pge/queryPRChunks.base
lib/xpl/pge/queryPRChunks.code
lib/xpl/pge/queryPRChunks.help
lib/xpl/pge/queryPRDetails.base
lib/xpl/pge/queryPRDetails.code
lib/xpl/pge/queryPRDetails.help
lib/xpl/pge/resubmitEpilogs.base
lib/xpl/pge/resubmitEpilogs.code
lib/xpl/pge/resubmitEpilogs.help
lib/xpl/pge/resubmitInstances.base
lib/xpl/pge/resubmitInstances.code
lib/xpl/pge/resubmitInstances.help
lib/xpl/pge/resumeInstances.base
lib/xpl/pge/resumeInstances.code
lib/xpl/pge/resumeInstances.help
lib/xpl/pge/searchJobs.base
lib/xpl/pge/searchJobs.code
lib/xpl/pge/searchJobs.help
lib/xpl/usr/
lib/xpl/usr/Makefile
lib/xpl/usr/isTokenValid.base
lib/xpl/usr/isTokenValid.code
lib/xpl/usr/isTokenValid.help
lib/xpl/usr/login.base
lib/xpl/usr/login.code
lib/xpl/usr/login.help
lib/xpl/usr/logout.base
lib/xpl/usr/logout.code
lib/xpl/usr/logout.help
log/
log/log.conf
test_suites/jira-cer-83/
test_suites/jira-cer-83/jira-cer-83-01.pl
test_suites/jira-cer-83/jira-cer-83-02.pl
test_suites/jira-cer-83/jira-cer-83-03.pl
test_suites/jira-cer-122/
test_suites/jira-cer-122/jira-cer-122-01.pl
test_suites/jira-cer-122/jira-cer-122-02.pl
test_suites/jira-cer-122/jira-cer-122-03.pl
test_suites/jira-cer-146/
test_suites/jira-cer-146/jira-cer-146-01.pl
test_suites/jira-cer-146/jira-cer-146-02.pl
test_suites/jira-cer-146/jira-cer-146-03.pl
test_suites/jira-cer-146/jira-cer-146-04.pl
test_suites/test_load_4_1_4_1P6_aqua.pl
test_suites/test_load_4_1_4_1P6_terra_2002.pl
test_suites/test_load_4_1_4_1P6_terra.pl
test_suites/test_load_4_1_4_2P4_aqua.pl
test_suites/test_load_4_1_4_2P4_terra_2002.pl
test_suites/test_load_4_1_4_2P4_terra.pl
test_suites/test_load_4_1_4_2P5_aqua.pl
test_suites/test_load_4_1_4_2P5_terra_2002.pl
test_suites/test_load_4_1_4_2P5_terra.pl
test_suites/test_load_4_1_4_3P3_aqua.pl
test_suites/test_load_4_1_4_3P3_terra_2002.pl
test_suites/test_load_4_1_4_3P3_terra.pl
test_suites/test_load_4_5_6_1P4_fm1_2002.pl
test_suites/test_load_4_5_6_1P4_fm1.pl
test_suites/test_load_4_5_6_1P4_fm2_2002.pl
test_suites/test_load_4_5_6_1P4_fm2.pl
test_suites/test_load_4_5_6_1P5_fm3.pl
test_suites/test_load_4_5_6_1P5_fm4.pl
test_suites/test_load_4_5_6_2P3_fm1_2002.pl
test_suites/test_load_4_5_6_2P3_fm1.pl
test_suites/test_load_4_5_6_2P3_fm2_2002.pl
test_suites/test_load_4_5_6_2P3_fm2.pl
test_suites/test_load_4_5_6_2P3_fm3.pl
test_suites/test_load_4_5_6_2P3_fm4.pl
test_suites/test_load_4_5_6_4P2_fm1_2002.pl
test_suites/test_load_4_5_6_4P2_fm1.pl
test_suites/test_load_4_5_6_4P2_fm2_2002.pl
test_suites/test_load_4_5_6_4P2_fm2.pl
test_suites/test_load_4_5_6_4P2_fm3.pl
test_suites/test_load_4_5_6_4P2_fm4.pl
test_suites/test_load_prs_2002_only.pl
test_suites/test_load_prs.pl

test_suites/test_login.pl
postgres_scripts/
postgres_scripts/logdb_clean_install
postgres_scripts/logdb_drop_tables
utilities/catalyst_logging_db_05032013.dump.bz2
utilities/db_ies_patch_20130812.txt
utilities/Missing_Aqua-FM3_Edition3_IES.txt
utilities/Missing_Aqua-FM4_Edition3_IES.txt
utilities/Missing_Terra-FM1_Edition3_IES.txt
utilities/Missing_Terra-FM2_Edition3_IES.txt
utilities/ies_db_patch.pl

Appendix E
Product Availability in CATALYST Database

E.1 CATALYST 1.0.4 Installation Package + Logging Database Patch (SCCR 996)

Based on the "as-delivered" static database contents listed below, CATALYST can successfully process the Edition4 Clouds/Inversion up to 2013-06-30. Attempting to process datadates beyond this will stall waiting for input; they will not run automatically producing non-optimal data. Static database patch deliveries will extend this date.

Product	Sampling Strategy	Production Strategy	Configuration Code	Starting Date	Ending Date
ESNOW/EICE	CERES	NSIDC-AFWA-Mesh16th	200200	20121230	20140223
ESNOW/EICE	CERES	NSIDC-CLASS	200200	20121211	20140223
ESNOW/EICE	CERES	NSIDC-NESDIS	020019	20000201	20021231
ESNOW/EICE	CERES	NSIDC-NESDIS	020018	20000301	20020930
ESNOW/EICE	CERES	NSIDC-NESDIS	021021	20030101	20050202
ESNOW/EICE	CERES	NSIDC-NESDIS	021022	20050201	20070131
ESNOW/EICE	CERES	NSIDC-NESDIS	021023	20070129	20090910
ESNOW/EICE	CERES	NSIDC-NESDIS	021024	20090831	20101031
ESNOW/EICE	CERES	NSIDC-NESDIS	022025	20101101	20130307
ESNOW/EICE	CERES	NSIDC-NESDIS-Mesh16th	022025	20081007	20130310
IES	Aqua-FM3	Edition3	032040	2002061800	2011010123
IES	Aqua-FM3	Edition3	300300	2010123100	2012100123
IES	Aqua-FM3	Edition3	300303	2012093000	2013063023
IES	Aqua-FM3	Edition3	300301	2012100100	2013022823
IES	Aqua-FM3	Edition4	400403	2002061800	2009123123
IES	Aqua-FM4	Ed3-NoSW	032040	2005033000	2011010123
IES	Aqua-FM4	Ed3-NoSW	300300	2010123100	2012100123
IES	Aqua-FM4	Ed3-NoSW	300303	2012093000	2013063023
IES	Aqua-FM4	Ed3-NoSW	300301	2012100100	2013022823
IES	Aqua-FM4	Ed4-NoSW	400403	2005033000	2009123123
IES	Aqua-FM4	Edition3	032040	2002061800	2005032923
IES	Aqua-FM4	Edition4	400403	2002061800	2005032923
IES	Terra-FM1	Edition3	032040	2000022900	2011010123
IES	Terra-FM1	Edition3	300300	2010123100	2012100123
IES	Terra-FM1	Edition3	300303	2012093000	2013063023
IES	Terra-FM1	Edition3	300301	2012100100	2013022823
IES	Terra-FM1	Edition4	400403	2000022900	2009123123
IES	Terra-FM2	Edition3	032040	2000022900	2011010123
IES	Terra-FM2	Edition3	300300	2010123100	2012100123
IES	Terra-FM2	Edition3	300303	2012093000	2013063023
IES	Terra-FM2	Edition3	300301	2012100100	2013022823
IES	Terra-FM2	Edition4	400403	2000022900	2009123123
MOA	CERES	DAO-G5-CERES	020032	2007100100	2010110100
MOA	CERES	DAO-G5-CERES	020033	2010110100	2012070118
MOA	CERES	DAO-G5-CERES	300300	2012063000	2014010118

Product	Sampling Strategy	Production Strategy	Configuration Code	Starting Date	Ending Date
MOA	CERES	DAO-G5re-CERES	019031	2007100100	2007103118
MOA	CERES	DAO-GEOS4	016024	1997121500	2004033018
MOA	CERES	DAO-GEOS4	016025	1998120100	2004120118
MOA	CERES	DAO-GEOS4	016023	2000022400	2004021518
MOA	CERES	DAO-GEOS4	016026	2004113000	2005010118
MOA	CERES	DAO-GEOS4	017027	2004123100	2005122918
MOA	CERES	DAO-GEOS4	017028	2005113000	2006020106
MOA	CERES	DAO-GEOS4	018029	2006020100	2007030118
MOA	CERES	DAO-GEOS4	018030	2007020100	2008010118
MOA	CERES	ECMWF-GEOS2	013016	1998010100	1998090118
MOA	CERES	ECMWF-GEOS3	015019	2000010100	2002080118
MOA	CERES	ECMWF-GEOS3	013016	2000022500	2000103118
MOA	CERES	ECMWF-GEOS3	014018	2000110100	2000111918
MOA	CERES	ECMWF-GEOS3	016020	2002080100	2002100112
MOA	CERES	ECMWF-GEOS4	016020	2002100100	2003043018
MOA	CERES	ECMWF-GEOS4	016021	2003050100	2003073118
MOA	CERES	ECMWF-GEOS4	016022	2003080100	2003093018
MOA	CERES	GMAO-G5-Beta2Ed3	300300	2004010100	2006013118
MOA	CERES	GMAO-G5-Beta2Ed3	300301	2006070100	2006073118
MOA	CERES	GMAO-G541-Ed4	400400	2000022800	2013070118
MOA	ValR2OS	DAO-GEOS4	018030	2006093000	2006110118
MOD02SS1	MODIS_TERRA	C5		200002240000	201312312355
MOD03	MODIS_TERRA	C5		200002240000	201312312355
MOD04_L2	MODIS_TERRA	C5		200002240010	201312312355
MYD02SS1	MODIS_AQUA	C5		200207032200	201312312355
MYD03	MODIS_AQUA	C5		200207030000	201312312355
MYD04_L2	MODIS_AQUA	C5		200207040045	201312312355
SCCD/SCCN	Aqua-FM3	Edition3	300301	200206	201304
SCCD/SCCN	Aqua-FM3	Edition3	300300	200207	201112
SCCD/SCCN	Aqua-FM3	Edition3	300303	201305	201306
SCCD/SCCN	Aqua-FM4	Ed3-NoSW	300300	200504	201112
SCCD/SCCN	Aqua-FM4	Ed3-NoSW	300301	201201	201304
SCCD/SCCN	Aqua-FM4	Ed3-NoSW	300303	201305	201306
SCCD/SCCN	Aqua-FM4	Edition3	300301	200206	200206
SCCD/SCCN	Aqua-FM4	Edition3	300300	200207	200606
SCCD/SCCN	Terra-FM1	Edition3	300301	200002	201304
SCCD/SCCN	Terra-FM1	Edition3	300300	200003	201112
SCCD/SCCN	Terra-FM1	Edition3	300303	201305	201306
SCCD/SCCN	Terra-FM1	Edition4	400403	200002	200912
SCCD/SCCN	Terra-FM2	Edition3	300301	200002	201304
SCCD/SCCN	Terra-FM2	Edition3	300300	200003	201112
SCCD/SCCN	Terra-FM2	Edition3	300303	201305	201306
SCCD/SCCN	Terra-FM2	Edition4	400403	200002	200912