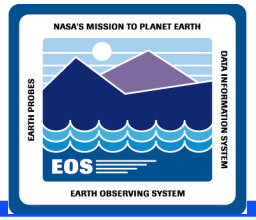
Extracting Prologs



The Prolog Extractor will search recursively for files with valid filename extensions. The beginning and end delimiters are:

!F77 !F90 !C !Ada !F77-INC !F90-INC !C-INC !PROLOG

!END

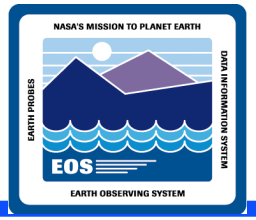
Key Procedure Commands:

- **SSIT Manager**
  - **Tools → Standards Checker → Prolog Extractor**

Run the Prolog Extractor GUI.

- **Select the directory with source files.**
- **Save or print the output Prologs files.**

Compiling and Linking Science Software

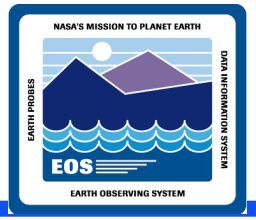


Science software developed at SCFs using the SDP Toolkit provided by ECS needs to be compiled and linked first with SCF Toolkit version to compare results at each facility. Then the science software needs to be compiled and linked with the DAAC Toolkit.

Preparation for compile and link:

- **Source correct SDP Toolkit library version - total of 8 versions.**
  - **Location Type: SCF or DAAC**
  - **Computer Language Type: FORTRAN 77, Fortran 90, C, Ada**
  - **Object Type: 32-bit mode or 64-bit mode, Old 32 bit (SCF's only)**
- **New and old 32-bit modes are distinguished by compiler options.**
- **Update PCF for execution of PGEs at the DAAC.**
- **Compile Status Message Files.**

Updating a PCF



PCF sections:

- **System Runtime Parameters**
- **Product Input**
- **Product Output**
- **Support Input**
- **Support Output**
- **User-defined Runtime Parameters**
- **Intermediate Input**
- **Intermediate Output**
- **Temporary I/O**

Update appropriate path names where necessary:

- **Add 10111|ShmMem|~/runtime||||1**

Updating a PCF

Checkout directory and file by typing:
cleartool checkout -nc .
cleartool checkout -nc *PCFFileName*

↓

Enter the vi editor by typing:
vi *PCFFileName*

↓

Edit the PCF.

↓

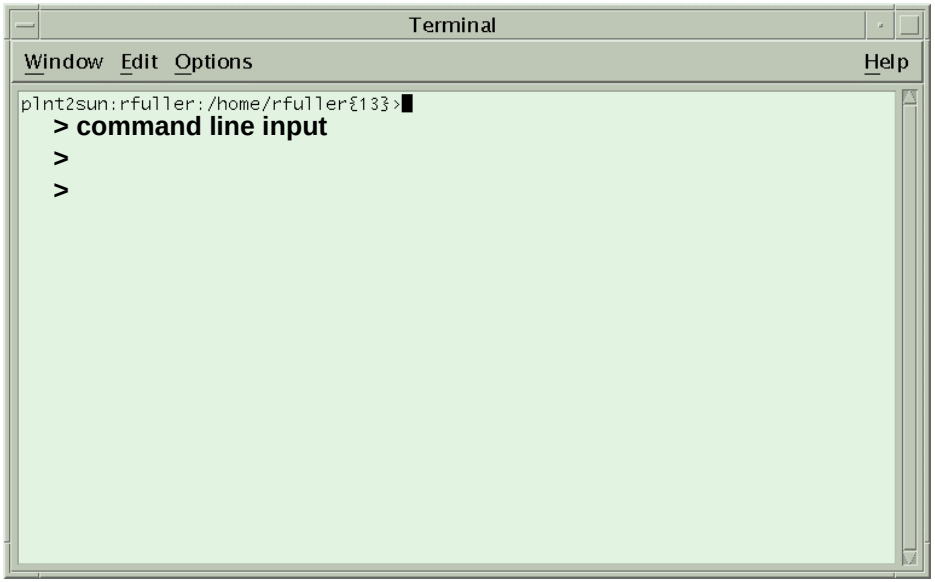
After editing, re-run the PCF Checker by selecting **Tools→Standards Checker→ PCF Checker** and from the SSIT menu.

↓

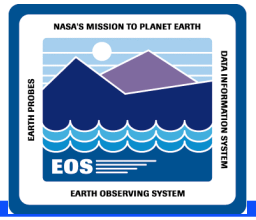
Checkin the file by typing:
cleartool checkin -nc *PCFFileName*

→

Checkin the directory by typing:
cleartool checkin -nc .



Compiling the SMF



Status Message Facility (SMF) Files - Also known as Error Status Message.

Provides:

- **An error and status message handling mechanism**
- **A method to send log files, informational messages and output data files to DAAC personnel or remote users.**

SMF files need to be compiled with science software into message files and include files.

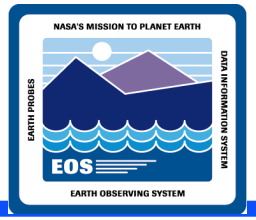
These files will be used by science software during runtime.

- **smfcompile -lang -f *TextFile.*t**
- **-lang is the computer language and *TextFile*.t is the SMF file.**

Process Steps:

- **Set ClearCase view (if source code is in ClearCase).**
- **Set up SDP Toolkit environment.**
- **Go to SMF directory for the PGE.**
- **Run the SMF compiler.**
- **Move created files to proper directories.**

Compiling a PGE and Linking with SCF and DAAC SDP Toolkits



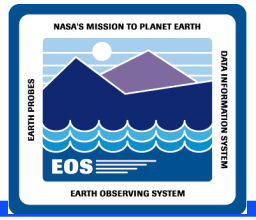
Compiling and Linking of Science Software will vary according to the instructions from the Instrument Software Development Teams.

Compiling and linking with SCF and DAAC versions differs only in the setting of the SDP Toolkit environment.

Process Steps:

- **Read all instructional information supplied with the delivery.**
- **Log into the SDPS SGI and set up the proper SDP Toolkit environment.**
- **Set the ClearCase view if software is already in ClearCase.**
- **Compile Status Message Facility files first.**
- **Examine the make or build file and alter if necessary.**
- **Using the make or build file, perform the build.**
- **If make file has been changed, check in modified version.**

Running a PGE in a Simulated SCF Environment



Running a PGE at the DAAC in a simulated SCF environment should produce identical results as those at the SCF.

Process Steps:

- **For SSI&T set up the SCF SDP Toolkit environment.**
- **For SSI&T Training set up the DAAC Toolkit environment (results will be the same).**
- **Set the environment variable PGS_PC_INFO_FILE to path and file name of PCF for the PGE.**
- **If the PGE has been run before in the same directory, remove old log files.**
- **Run the PGE from the command line.**

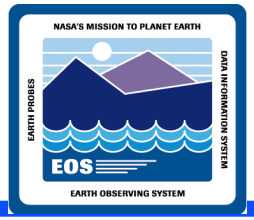
Running a PGE in a Simulated SCF Environment (cont.)

To capture PGE runtime statistics for the PDPS Database, perform profiling using the DpPrRusage Program (Rusage).

Statistics needed:

- **wall clock time**
- **user time**
- **system time**
- **amount of memory used**
- **number of page faults**
- **number of input and output blocks**
- **number of swaps**

Running a PGE in a Simulated SCF Environment (cont.)



Sample of an Rusage File produced:

lasher{emcleod}6: more Profile.out

source .cshrc

 # cd TEST/MOD*

 # ls

#

/usr/ecs/OPS/CUSTOM/bin/DPS/EcDpPrRusage MOD_PR10.exe > Profile.out

lasher{emcleod}9: more profile.out

 # Resource Usage Information

COMMAND=MOD_PR10.exe

EXIT_STATUS=0

ELAPSED_TIME=233.583145

USER_TIME=10.046158

SYSTEM_TIME=7.555547

MAXIMUM_RESIDENT_SET_SIZE=4080

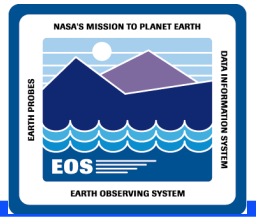
AVERAGE_SHARED_TEXT_SIZE=0

AVERAGE_UNSHARED_DATA_SIZE=0

AVERAGE_UNSHARED_STACK_SIZE=0

- **PAGE_RECLAIMS=151**
- **PAGE_FAULTS=0**
- **SWAPS=0**
- **BLOCK_INPUT_OPERATIONS=2**
- **BLOCK_OUTPUT_OPERATIONS=2710**
- **MESSAGES_SENT=0**
- **MESSAGES_RECEIVED=0**
- **SIGNALS_RECEIVED=0**
- **VOLUNTARY_CONTEXT_SWITCHES=1095**
- **INVOLUNTARY_CONTEXT_SWITCHES=2**
- **lasher{emcleod}10:**

Examining PGE Produced Log Files



PGEs produce three log files during runtime:

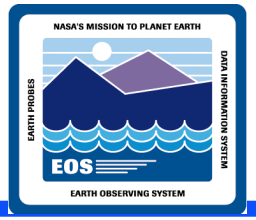
- **Status Log - captures all error and status information.**
- **User Log - captures a subset of more informational messages.**
- **Report Log - captures arbitrary message strings.**

Log file messages are written by both SDP Toolkit and science software using the Status Message Facility (SMF).

Process Steps:

- **Examine PCF to get location of log files.**
- **With SCF version of Toolkit, location and filenames can be set as desired.**
- **Look for errors or warnings, anomalous messages**

File Comparison and Data Visualization



An important activity for SSI&T is comparing the output data products from the PGE runs to test files delivered with the PGE.

The comparison may consist of display of metadata in HDF files, display of differences in data values, or display of images of the data products.

Searches are performed for any differences beyond specified tolerances.

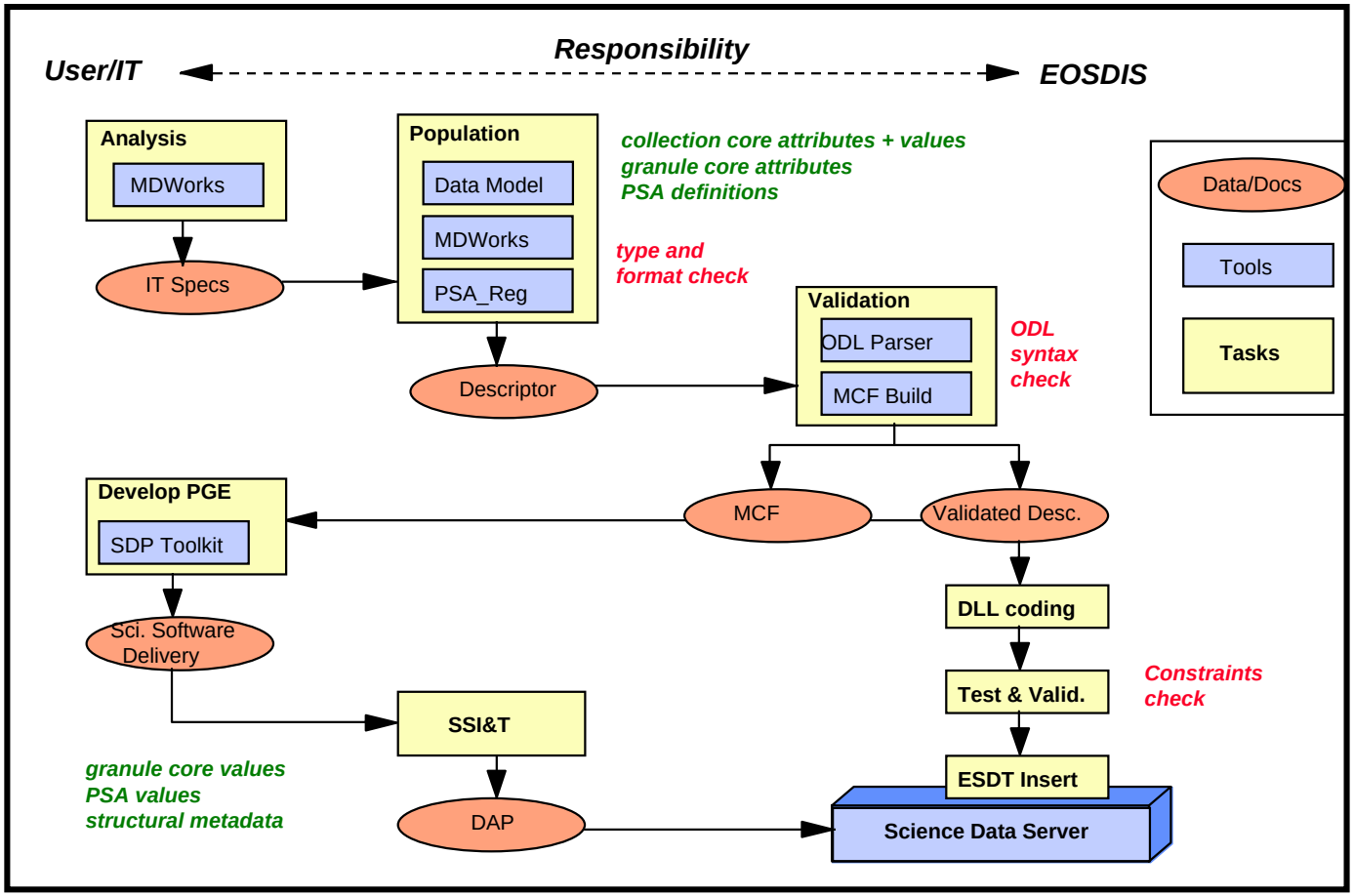
Data product files can be compared by a variety of tools accessible by the SSIT Manager GUI.

- **Tools → Product Examination → File Comparison → HDF or ASCII or Binary**

Data visualization tools are accessible by the SSIT Manager GUI.

- **Tools → Product Examination → IDL or EOSView**

The Metadata Workflow



Earth Science Data Types- ESDT's

- **Representation of different types of data products from the scientists perspective.**

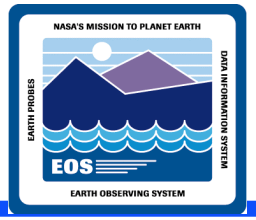
- **Define to the Version 2 Data Model:**
  - **Collection level metadata attributes and values.**
  - **Granule level metadata attributes.**
  - **Data services appropriate to the ESDT collection and data granules within the collection.**

ESDT Components

Descriptor File:

- **Collection level metadata attributes and values.**
- **Granule metadata attributes.**
- **Granule metadata attributes valid values.**
- **Services to be performed for the science data.**
- **The set of attributes in the granule level part of the descriptor is the source for producing a Metadata Configuration File (MCF). From the MCF will be produced a .met file using the SDSRV.**
- **DLL File:**
- **The Dynamic Link Libraries (DLL) file is uniquely produced for each ESDT and must be installed with each ESDT.**

Preparation of Earth Science Data Types (ESDTs)

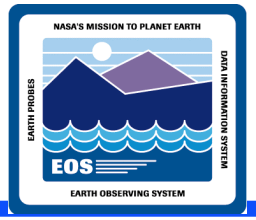


Building and installing ESDTs has been done before ECS is applied.

ECS Requirements:

- **ESDTs for all data collections to be input to PGEs or output from PGEs must be built and registered into ECS (SDSRV) before any PGEs are run in PDPS.**
- **Version 2 uses Science Dataserver Operator GUI to install ESDT's.**
- **Reasons for Inclusion in SSI&T:**
- **Instrument Teams may deliver new ESDTs for new types of input files and output products from PGEs.**
- **NCR process for updates/changes to ESDT Descriptors.**
- **Some Ancillary input ESDT's may be created by the DAAC's in the near term.**

MCF Generation from SDSRV



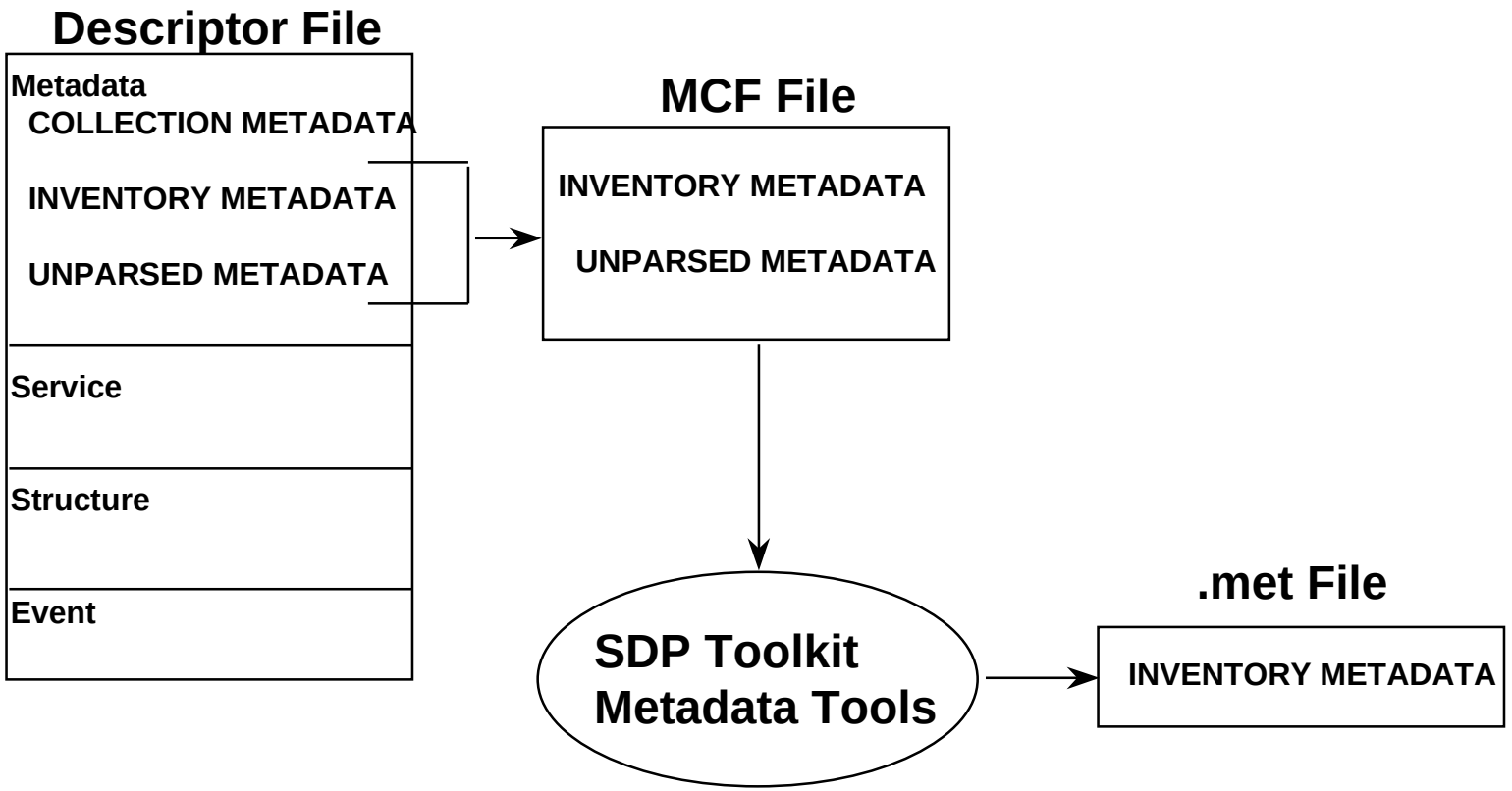
The Metadata Configuratrion File (MCF) is produced by the following processes:

- **The ESDT's and DLL's are installed into the SDSRV with error checking taking place in the descriptor before installation.**

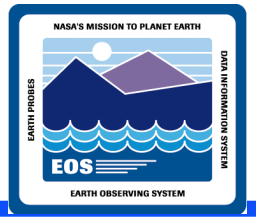
- **The GETMCF tool is executed to pull the MCF from the SDSRV. The actual MCF is generated and then copied from the Inventory and Archive section of the Collection Descriptor.**

-

Descriptor, MCF & .met Files



What are Valids?



- **Valids are type of integrity constraint to ensure that metadata values comply with the data model and database schema requirements**
- **Valids are used to ensure the data products quality and consistency of search queries**
- **Currently implemented for**
  - **Data Type and Length checking; *e.g. STRING, FLOAT, etc.***
  - **Match Rules; *e.g. DayNightFlag = (Day, Night, Both) ex: DAY will not match, resulting in a failed insert. (match rule is exacting)***
  - **Range Checking ; *e.g.***

    Longitude Minimum = -180.00 to

    Longitude Maximum = +180.00
  - **Expressions; e.g. *MinimumAltitude >= 0.0***

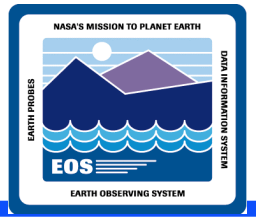
Attribute Valid Processing

- Attribute Valids have dependency with corresponding Descriptors, PGEs; Versions must be “in-sync”
  - Attribute Valids values may have a dependency upon SDSRV version and potentially other subsystems(Client & Ingest for example)
  - Changes to Valids may impact code; Versions must be “in-sync”
  - Attribute Valids are stored within SDSRV metadata database
- Incoming metadata validation processing is imbedded within the SDSRV software
  - Action on ESDT/granule with invalid metadata depends on “MANDATORY=“ setting

Attribute Valid Processing (Cont'd)

- Currently, incoming metadata checked against one of the 4 basic constraints checks (Match, Range, Expression, or NONE) as specified
- "NONE" means no value checking is performed
- Messages are logged for attributes containing invalid data. Action on ESDT/granule with invalid metadata depends on "MANDATORY=" setting

PSA Process

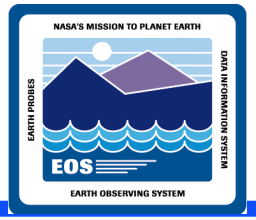


- Product Specific Attributes (PSA) information is obtained from Instrument Teams through a template (PSA Template) that has been provided to them
- Data Engineering performs analysis to verify that the PSAs are unique and conform to the Data Model
- Approved PSAs are submitted to the ECS CCB for approval to update the PSA baseline
- PSA_Registry database is updated with the new approved PSAs
- Reports are generated on a bi-weekly basis or on as-needed-basis
- PSA reports are posted on the ESDT Bulletin Board (internal) and on the ECS Web Page
  - URL = http://ecsinfo.hitc.com/metadata/psatables.html

ESDT CM Process

- Stored in ClearCase
- Directory structure is established to differentiate the different versions of ESDTs
  - Allows support of multiple versions of Database valids and schema
  - Allows for the creation of custom ESDT changes in order to support the evolution of code development
  - Makes delivery to different sites/platforms (mini-DAAC, VATC, GSFC, etc…) easier
- Changes to ESDTs are based upon input from development, and ITs
- Modified ESDTs are merged onto the baseline after approval at the merge meetings

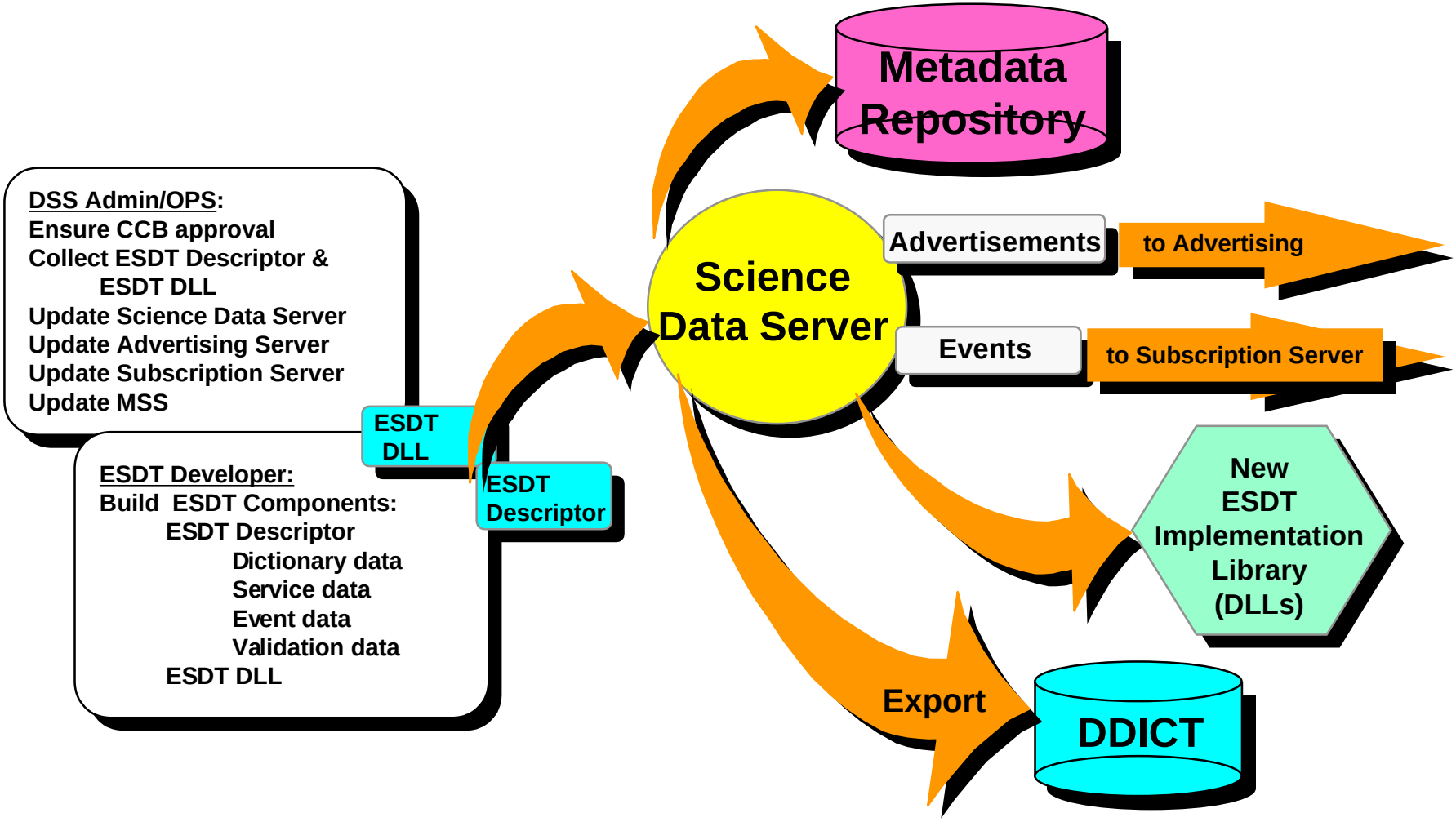
Overview: ESDT Development and Installation Process



Complete ESDT development and installation process involves the following steps:

- **ESDT Generation**
- **ESDT Installation in the SDSRV**
  - **Both the Descriptor and DLL are stored within the Science Data Server**
  - **From the Science Data Server, the attribute information contained in the ESDT Descriptor is passed into a number of Clients as depicted on the next slide (Adding a new ESDT)**

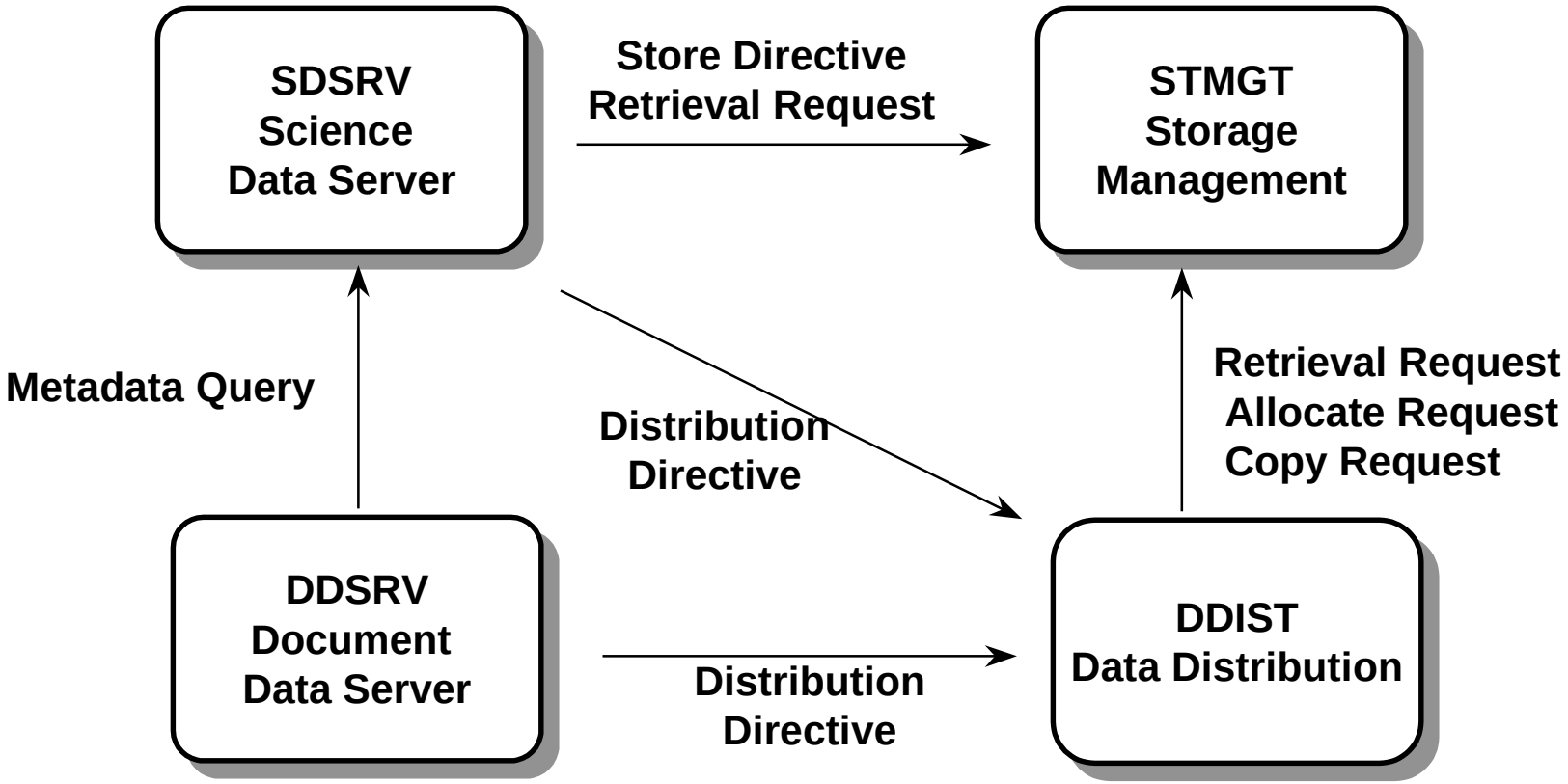
Adding a new ESDT: Operational Overview



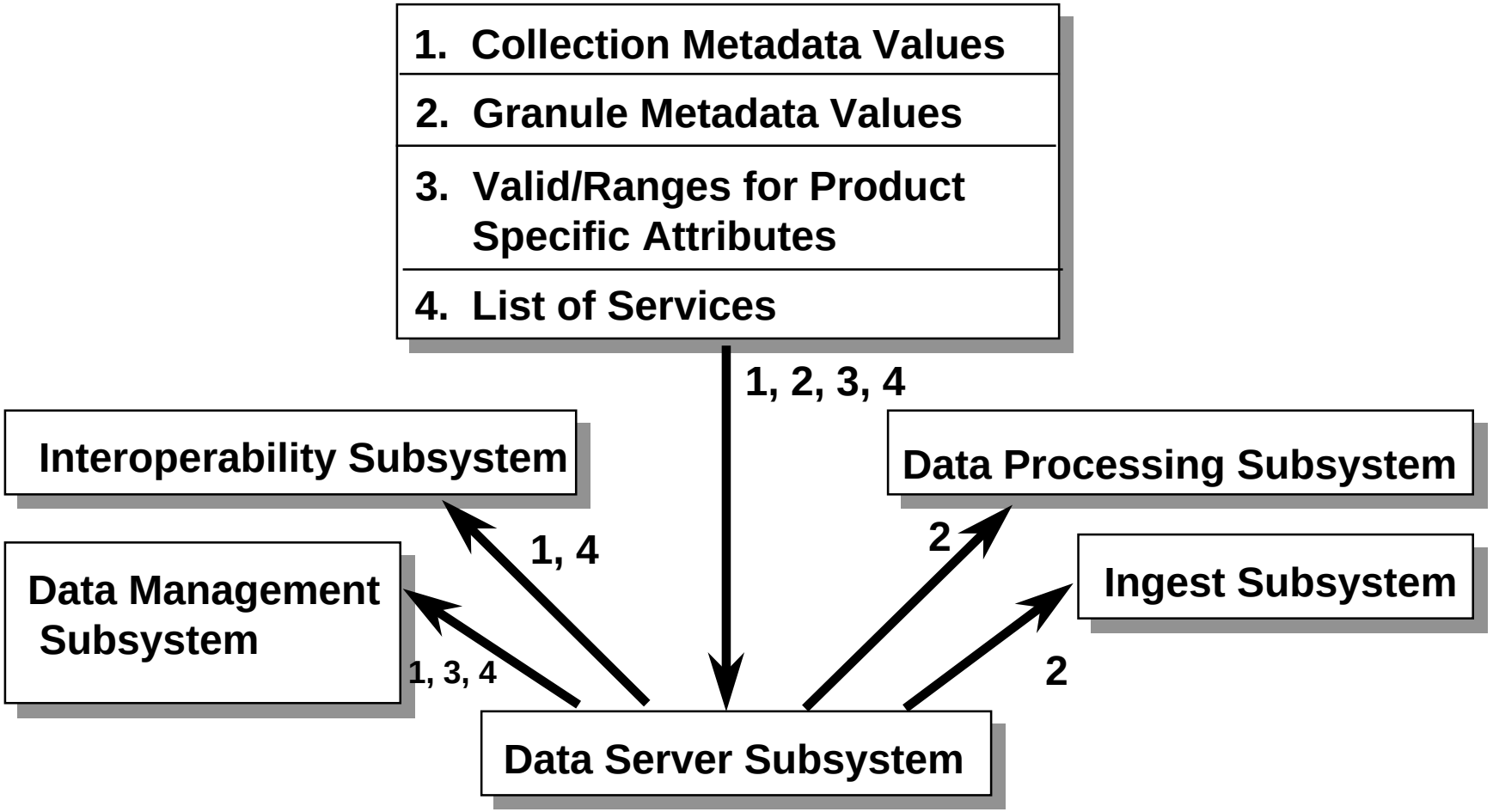
Required Servers

- **Following servers need to be started and running before installing ESDTs (with GDAAC machine names as examples):**
  - **Science Data Server (SDSRV) (g0acs03)**
  - **Storage Management Servers (STMGT) (g0icg01, g0drg01, g0dps02)**
  - **Data Distribution Servers (DDIST)(g0dps02)**
  - **Subscription Server (IDG) (g0ins01)**
  - **Advertising Server (IOS) (g0ins02)**
  - **Data Dictionary Server (IOS) (g0ins02)**
- **ESDTs (both components -- descriptor and corresponding DLL files) to be installed must exist and must have been verified for syntax, valids, and other metadata attributes correctness**

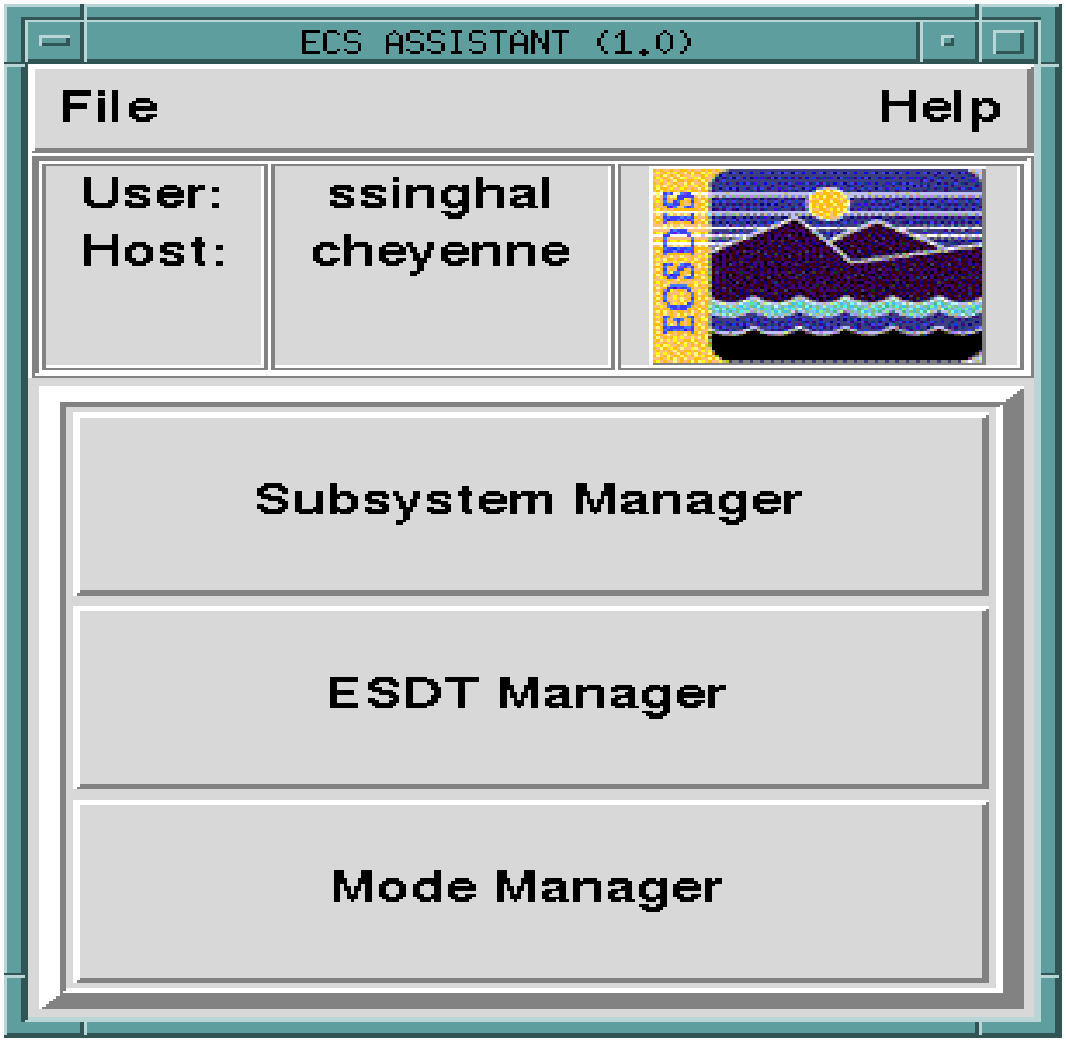
Data Server Subsystem: Software Components



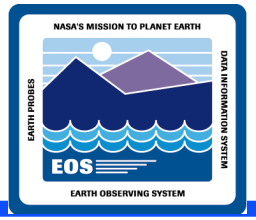
Metadata Flow in ECS



Tools - EcCoAssist



Bringing Up ECS Assistant



To Bring up ECS Assistant, execute the procedure steps that follow:

1 **At the UNIX Console or Terminal type setenv DISPLAY ……:0.0, To verify the setting, type echo $DISPLAY, press Enter.**

2 **Create an xterm by typing: xterm -n hostname &**

The hostname is the name of the machine on which the ECS Assistant is to be displayed, i.e., the machine that your are using.

3 **Log into one of the host machines used for SSIT, (Tested using telnet p0acs03), ID:, PASSWORD:.**

Bringing Up ECS Assistant continued

4 If necessary, at the UNIX prompt on the host from which the ECS Assistant is to be run, type cleartool setview ViewName, press Enter.

The ViewName is the ClearCase view to be used while the ECS Assistant is running in this session. For example, type cleartool jdoe, press Enter.

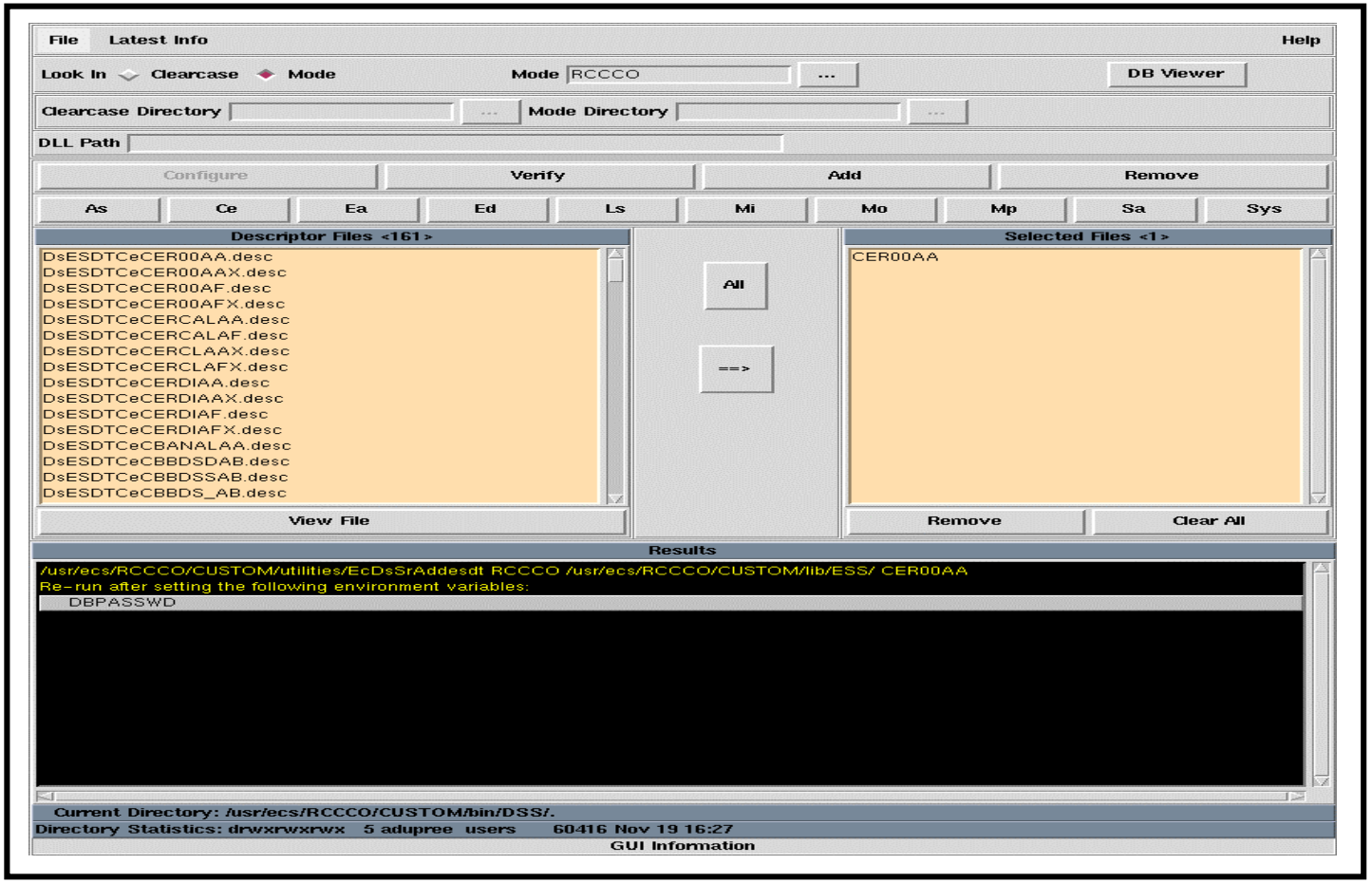
A ClearCase view is required only if the ECS Assistant needs to be able to "see" into a ClearCase VOB; a view is not necessary otherwise.

5 At the UNIX prompt, type cd /tools/common/ea, press Enter. Then type: EA, press Enter.

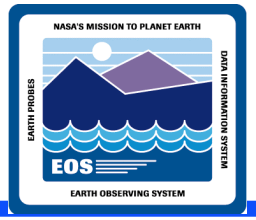
- /tools/common/ea is the path where ECS Assistant is installed.

This will invoke the ECS Assistant GUI with three push buttons for selecting the proper activities, as indicated in the previous picture.

ESDT Manager GUI



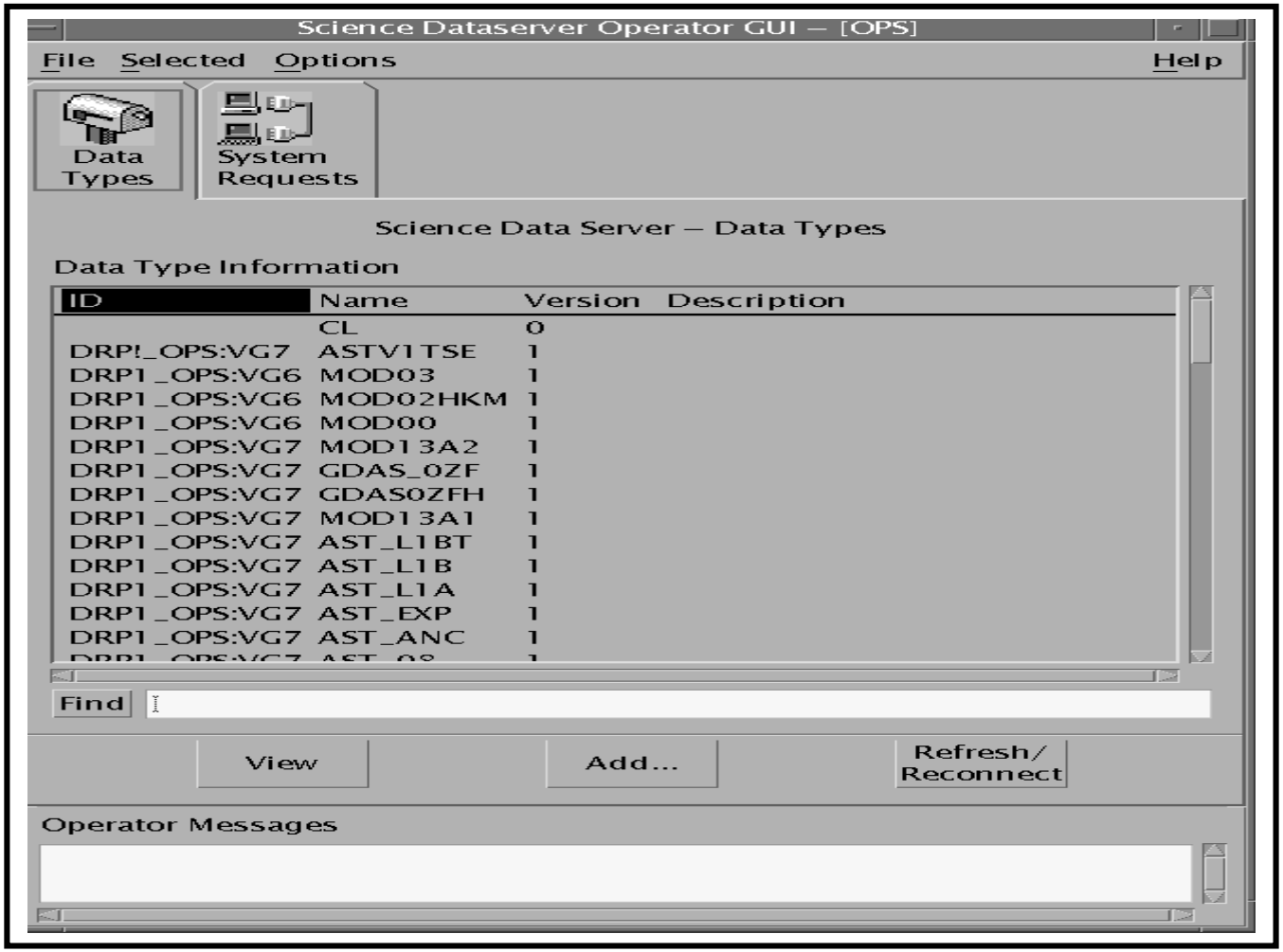
Installing an ESDT/DLL using Science Data Server Operator GUI



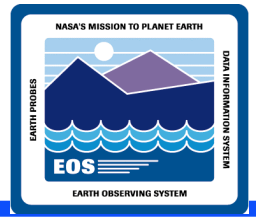
Key Assumptions

- **Your are logged into the SDSRV with the necessary servers listening.**
  - **SSI&T personnel have permissions and privileges to register ESDTs.**
  - **Example: telnet p0acs03, login ID:, password:, setenv DISPLAY.**
  - **Then log into: dce_login awhitele awhitele , setenv DISPLAY**
  - **cd /usr/ecs/<mode>/CUSTOM/utilities/EcDsSdsrvGuiStart <mode> .**
  - **The SDSRV Operator GUI should now appear.**
  - **The ESDT's and universal DLL descriptor file are normally installed dynamically as one when the ESDT is registered. They also can be installed from a specific mode or by first copying into the selected mode those ESDT with a compatable DLL**
  - **required for a particular PGE. Example:**
  - **DLL located: /home/emcleod/ESDT**
  - **ESDT Descriptors Located: /home/emcleod/ESDT**

Installing an ESDT/DLL using Science Data Server Operator GUI



Installing an ESDT/DLL using Science Data Server Operator GUI continued



Entering from Command Line

EcDsSdsrvGuiStart <MODE>. This will bring up a GUI where you can click on ADD. Another GUI will appear "Add Data Type". Enter the following:
Descriptor Filename: Enter path to where ESDT is located, including the full ESDT descriptor.

Archive ID: enter DRP1_OPS
click on: OK
If added successfully, another GUI will appear saying …DataType Successfully Added.

Verify installation by looking at the log: path is - cd /usr/ecs/CUSTOM/logs, type in: ls -lrt, to get the latest ALOG entry. type in: more EcDsScienceDataServer.ALOG to display same.

Viewing and Copying ESDT/DLL using ECS Assistant GUI

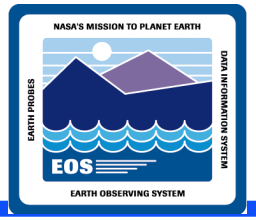
Key Assumptions

- **The ECS Assistant is up with the necessary servers listening.**
  - **SSI&T personnel have permissions and privileges to register ESDTs.**
  - **The ECS Assistant GUI is running with ESDT Manager selected.**
  - **The ESDT and DLL descriptor files are installed in the specific mode.**
  -
  - **ESDT Descriptors Located:**
  - **/usr/ecs/TS1/CUSTOM/data/ESS**

    DLL located: /usr/ecs/TS1/CUSTOM/lib/ESS

The ESDT's and universal DLL descriptor file are normally installed dynamically as one when the ESDT is registered using the Science Dataserver Operator GUI.

Validating Successful ESDT Installation



Criteria for success

- **The SDSRV will display an event ID to the fact that: MM/DD/YY HH/MM Finished adding ESDT.**
- **The following servers will also need to have acknowledged a successful ESDT Event ID before additional work can be done:**
- **ADSRV, DDICT, & SBSRV.**

Removing ESDT's Using the Command Line

Removing ESDT's from Archive Area using Command Line

Procedures:

1 **telnet** to (SDSRV) **p0acs03**[e.g.]

2 login: **cmts1**, password: **ecsu$er**

3 Login to DCE **(dce_login <name> <Password>**), **setenv DISPLAY ….:0.0**

4 **cd dbr**

5 **source dx.csh**

%dbr

First delete ESDT"s from the Advertisement Subsystem:

****

Removing ESDT's Using the Command Line continued

6 rlogin p0ins02 -l cmts1

7 *Login to DCE* **(dce_login <name> <Password>*), setenv DISPLAY ….:0.0***

8 rlogin p0ins02 -l ios

9 *Login to DCE* **(dce_login <name> <Password>*), setenv DISPLAY ….:0.0***

10 cd /usr/ecs/OPS/CUSTOM/utilities

11 setenv MODE OPS

12 source EcCoEnvCsh

13 cd /usr/ecs/OPS/CUSTOM/bin/IOS

ContributionDriver OPS

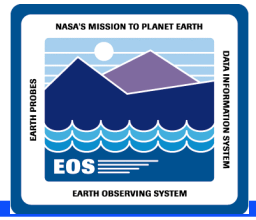
awhitele

awhitele

3

2

Removing ESDT's Using the Command Line continued



14 your_short_name_here

- y

Success is when the "<" prompt returns

To make sure the advertisements are deleted from the database

15 **p0ins02**% isql -Uios_role -Pwelcome –S**p0ins02**_srvr

- [If not OPS mode]
- 1> use IoAdAdvService_MODE

[where MODE is your mode, e.g. TS1]

- [if OPS mode}

1> use IoAdAdvService

2> go

1> select * from IoAdAdvMaster where title like "%your_short_name_here%"

2> go

Result should be no rows returned.

Removing ESDT's Using the Command Line continued

If you do get rows returned, the delete from advertisement did not work.

****

Then delete ESDT

****

16 rlogin [**p0acs03**] -l **id**, **pw**:

17 *Login to DCE* **(dce_login <name> <Password>)**, ***setenv DISPLAY ….:0.0***

18 cd /usr/ecs/OPS/CUSTOM/utilities

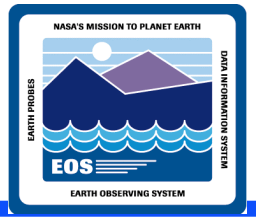
19 EcDsSrRmesdt OPS your_short_name_here

Success is no error msgs

****

Kill servers -- AFTER WARNING EVERYONE WORKING IN YOUR MODE!

Removing ESDT's Using the Command Line continued



****

20 Using ECS Assistant : # kill Sdsrv & HdfEosSrv & AdSrvr

****

#This will clean up DCE 's. Using ECS Assistant : # Restart serversl Sdsrv & HdfEosSrv & AdSrvr

****

- # start Sdsrv & HdfEosSrv on [**p0acs03**]
- # start SubSrvr on [**p0ins02**]

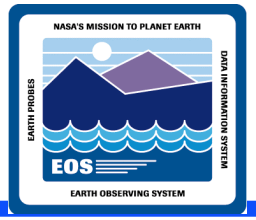
(cleanup done automatically)

- sdsrv.startup OPS
- ios-dm-mss.startup OPS

****

Now reinstall the ESDT on SDSRV

Production Rules Syntax Overview



PGE Registration

PGE Registration ODL format

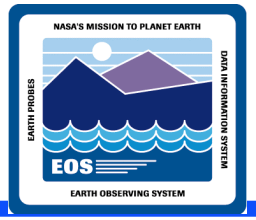
Production Rules

Rule Descriptions and Corresponding Syntax

Final Notes

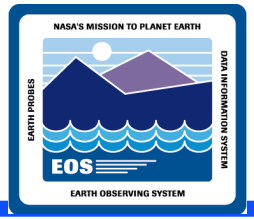
Science Software and Production Request

PGE Registration



- PGE Registration is the step during Science Software Integration and Test (SSIT) that defines a PGE to PDPS.
- Information about the PGE is put into ODL (see next slide) files for ingestion into PDPS.
- These files are read by SSIT software. The data describing the PGE is stored in the PDPS database.
- When the PGE is executed in the production environment, the information in the PDPS database is retrieved to schedule the PGE for execution. The information tells PDPS when the PGE should be scheduled, what data needs to be present to run the PGE, and what processing resources are needed.
- This will be done by the SSIT Operators at the DAAC. The Instrument Teams may be asked to fill out a web page that describes their PGE so that the SSIT operators can properly create the ODL files.

PGE Registration ODL format



- **ODL is simply a parameter=value file format. Each line in the file (except for comments) is of the form: parameter = "value"**

  There are currently 5 types of ODL files that need to be specified for PGE registration.

  - **The first defines the PGE itself and is called the PGE metadata. This contains information about the PGE, its name, instrument, schedule type, as well as definitions of its inputs and outputs.**
  - **The second defines any ESDTs that the PGE uses (as input or output) and is called the ESDT metadata. This describes the data that a PGE will use. Note that there will be one of these files for every ESDT used by the PGE.**
  - **The third defines the Tile definitions. It is only needed for those PGEs who are "Tile Scheduled".**
  - **The fourth defines the Orbit times for the platform of the instrument for which the PGE is run. This is only required if the PGE wants to run on during specific orbits of the spacecraft or process data based on those orbits.**
  - **The fifth defines the mapping of path numbers. It is only required for PGEs that require path numbers for runtime parameter values**

Production Requests

Science Software and Production Requests

Science software is one of the keys to production planning and processing:

- **Performs the actual data processing to create desired products.**
- **Is developed at Science Computing Facilities (SCFs) external to ECS.**
- **Is embodied in Product Generation Executives (PGEs) when the software is integrated into the ECS production processing environment.**
  - **PGEs are science software code (e.g., executable programs or shell scripts) that contain the instructions for processing data to create the desired products.**

Production Request Continued

The production request (PR) is another key to production planning and processing. The Production Planner defines ECS science data processing in terms of PRs.

- **A PR is an order for data to be produced by the Data Processing Subsystem.**
- **A single PR may specify several jobs (using the same PGE) that are to be run over a period of time or a single job producing a single set of data.**
- **PRs may apply to the processing of new data (standard PRs or standing orders) or the reprocessing of existing data (reprocessing PRs).**
- **Each PR identifies a specific PGE for generating a particular type of product.**
  - **Some PGEs are dependent on others; i.e., some PGEs require input data that are the output of other PGEs.**
  - **The planning software will recognize and reject a PR when the PR specifies a PGE that requires data from another PGE that has not yet been specified in a PR.**

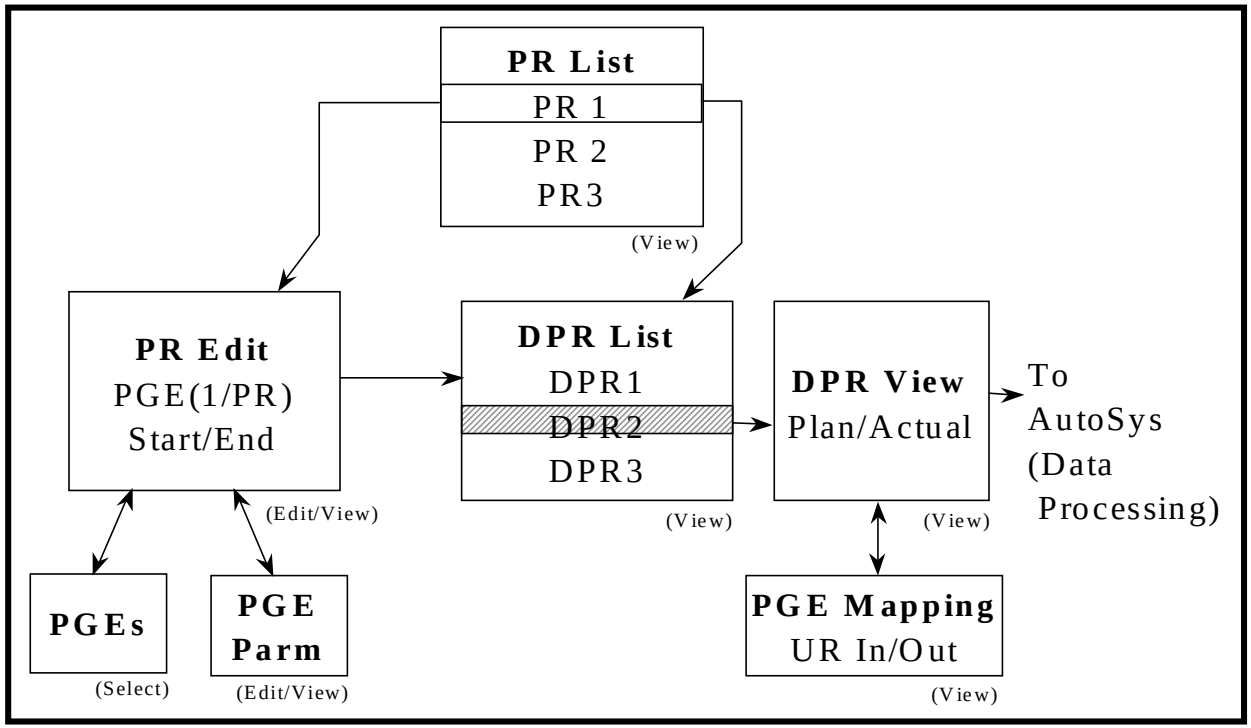
Production Request Continued

The Planning Subsystem performs the following functions:

- **Uses each PR to generate either one or a series of Data Processing Requests (DPRs).**
  - **Each DPR corresponds to one execution of a single PGE.**
  - **Each DPR contains the information that is needed by the SDP processing function, including PGE-related information.**
- **Checks the availability of the data required for the DPR, either from the Science Data Server (if the data have been previously ingested) or from internal predictions (if the data are expected to arrive in the future).**
- **Determines what data will be included in the DPR output so the system can make predictions concerning the availability of data for subsequent PGEs.**

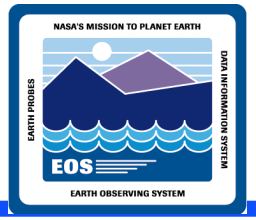
Production Request Continued

This figure shows the relationships among the PGEs, PRs, and DPRs as they are accessed through the Production Request Editor GUI.



Production Request Editor Flow

Production Request Continued



Types of Processing

ECS either accommodates or will accommodate the following four general types of data processing:

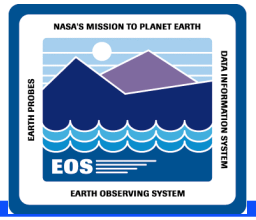
- **Routine Processing**
- **Reprocessing**
- **Ad-Hoc Reprocessing**
- **On-Demand Processing**

Production Rules

Provides a template for Instrument Teams to describe the relationship(s) between the PGEs and the input and output data. These specifications cover a variety of issues such as:

- **Basic Temporal specification**
- **Advanced Temporal specification**
- **Alternate Inputs**
- **Optional Inputs**
- **Minimum/Maximum Number of Granules**
- **Optional DPRs**
- **Intermittent Activation -Every nth DPR is activated; all others skipped**
- **Metadata Checks-based Conditional PGE activation**
- **Metadata Query**
- **Data Day**

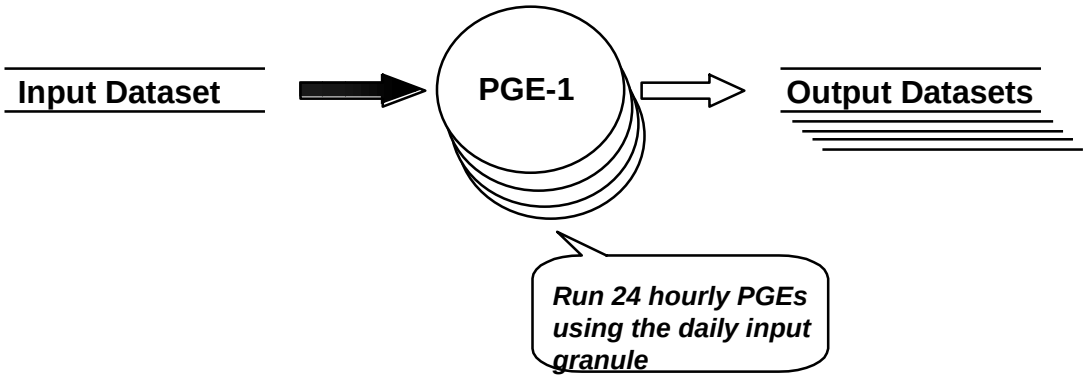
Production Rules Continued



- Spatial Query
- Tiling
- Closest Granule
- Orbital Processing
- PGE Exit Conditions

Basic Temporal

Specify temporal range of inputs that matches temporal range of outputs.



Production Rule Information Needed for PGE Registration:

- **PGE Schedule Type is "Time Scheduled"**
- **Boundary and Period of PGE are specified.**

“Time Scheduled” ODL Files and Parameters

The following information needs to be filled out in the PGE metadata file:

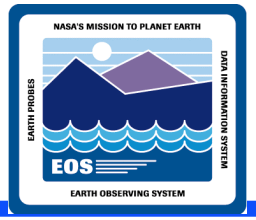
- **Schedule_Type = “Time”**
- **Processing_Boundary = <time boundary on which PGE runs>**
- **Processing_Period = <interval between PGE runs>**

The Boundary and Period will normally match the expected length and start time of the input data for the PGE. For a PGE that runs on 24 hours worth of data, the Boundary/Period would be set to:

Processing_Boundary = “START_OF_DAY”

- **Processing_Period = “DAYS=1”**

“Time Scheduled” ODL Files and Parameters (CONT)



The following information needs to be filled out in the ESDT metadata file (for each ESDT that comes from an external source):

- **Boundary = <time boundary on which data is collected>**
- **Period = <interval of data collection>**

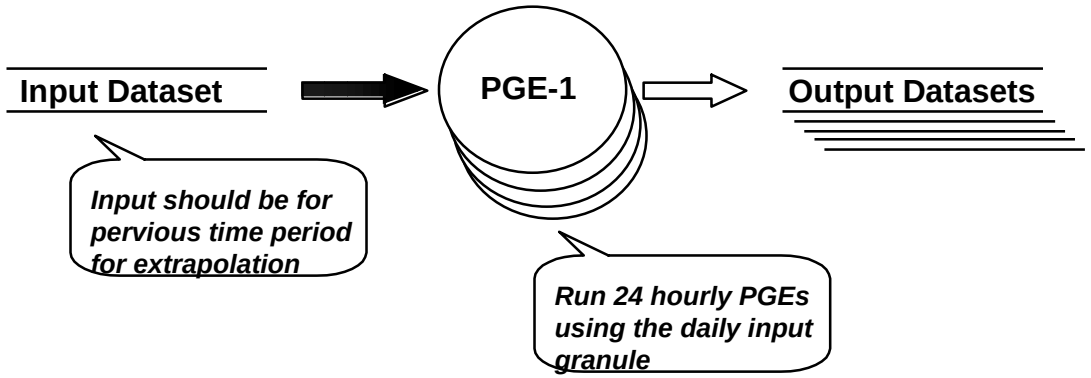
Data from an external source is periodic data that is not produced by PGEs at the local DAAC. For data that comes in every 2 hours (such as EDOS data):

Processing_Boundary = “START_OF_DAY”

- **Processing_Period = “HOURS=2”**

Advanced Temporal

Specify temporal range of inputs with offsets from expected temporal range of inputs and outputs.



Production Rule Information Needed for PGE Registration:

- **PGE Schedule Type is "Time Scheduled"**
- **Boundary and Period of PGE are specified as if Basic Temporal.**
- **Offsets are specified for data that is to be retrieved for another time period.**

“Advanced Temporal” ODL Files and Parameters

The following information needs to be filled out in the PGE metadata file for each input Data Type (PCF Entry) that is to have a timeframe different from the PGEs output:

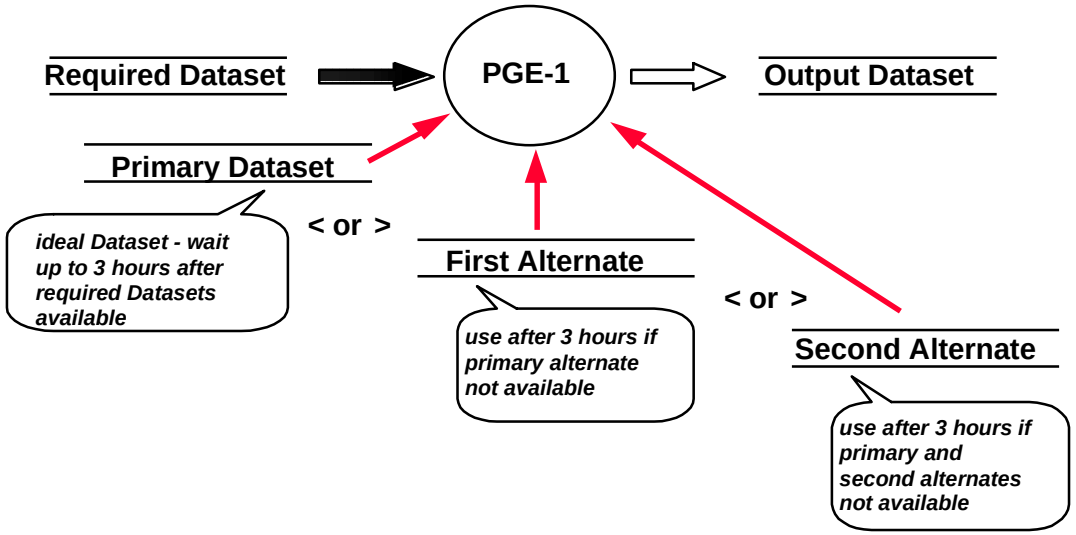
- **Begin_Period_Offset = <number of seconds to add (+) or subtract (-) from the start collection time when requesting data>**
- **End_Period_Offset = <number of seconds to add (+) or subtract (-) from the end collection time when requesting data>**

If the PGE runs every hour and desires data for the pervious hour and the current hour (2 granules):

- **Begin_Period_Offset = -3600**
- **Processing_Period =0**

Alternate Inputs

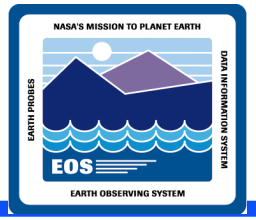
Run PGEs with different inputs based on availability or quality of various alternate input data sets.



Production Rule Information Needed for PGE Registration:

- **Schedule Type and corresponding data.**
- **Alternate Input Objects for inputs that have alternates.**

"Alternate Input" ODL Files and Parameters



The following information needs to be filled out in the PGE metadata file for each Data Type (PCF Entry) that is or has Alternate Inputs:

- **Input_Type = "Primary"**

(This is the Primary -- First Choice -- Alternate)

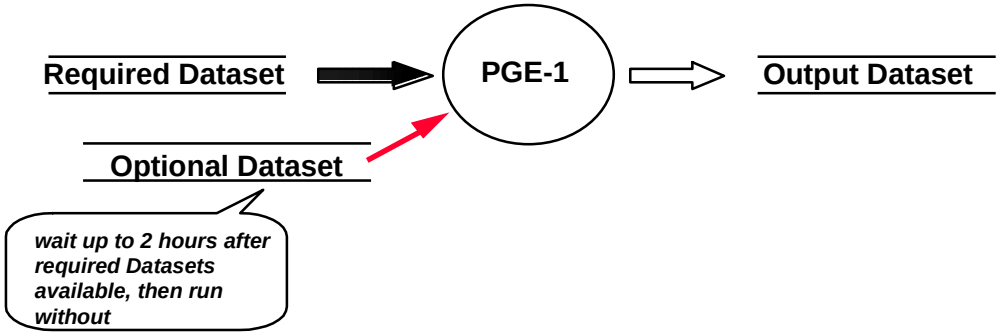
- **Input_Type = "Alternate"**

(This is for every other Alternate)

- **Object = Alternate_Input**
- **Category = <name of list of alternates; same for every alternate in the list>**
- **Order = <number indicating which alternate comes first, second,…>**
- **Timer = <number of days/hours/seconds to wait for alternate>**
- **WaitFor = <should we wait for this alternate; should be false for all but the last alternate in the list>**
- **Temporal = <does this alternate data type have temporal component>**
- **End Object = Alternate_Input**

Optional Inputs

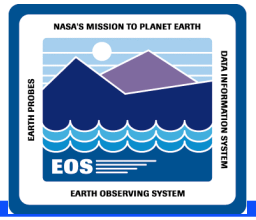
Run PGE with specified inputs if available; otherwise run PGE without them.



Production Rule Information Needed for PGE Registration:

- **Schedule Type and corresponding data.**
- **Optional Input Objects for inputs that are optional.**

“Optional Input” ODL Files and Parameters

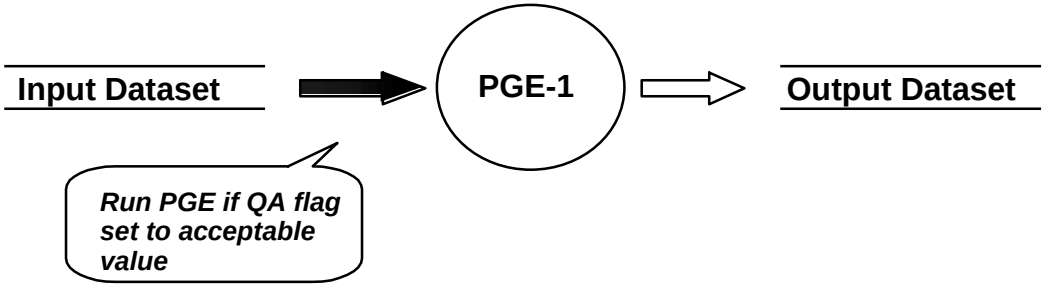


The following information needs to be filled out in the PGE metadata file for each Data Type (PCF Entry) that is an Optional Input:

- **Input_Type = “Optional”**
- **Object = Optional_Input**
- **Category = <name of list of optionals; same for every alternate in the list>**
- **Order = <number indicating which optional comes first, second,…>**
- **Timer = <number of days/hours/seconds to wait for optional>**
- **Temporal = <does this optional data type have temporal component>**
- **End Object = Optional_Input**

Metadata-based PGE Activation

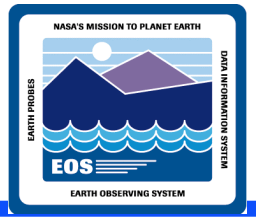
Use metadata of input data set to determine whether a given PGE is to be run.



Production Rule Information Needed for PGE Registration:

- **Schedule Type and corresponding data.**
- **Metadata Checks Objects for inputs to be checked.**
- **Metadata Definition objects specifying the type of metadata parameter.**

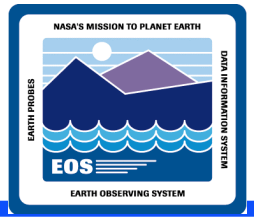
“Metadata Activation” ODL Files and Parameters



The following information needs to be filled out in the PGE metadata file for each input with a metadata check:

- **Object = Metadata_Checks**
- **Parm_Name = <name of metadata parameter to check>**
- **Operator = <operator on parameter value>**
- **Value = <value to be checked against>**
- **Database_Query = <get value from PDPS database; valid values are: NONE, ORBIT NUMBER, PATH NUMBER, TILE ID, START DATA DAY END DATA DAY>**
- **End Object = Metadata_Checks**

“Metadata Activation” ODL Files and Parameters (CONT)

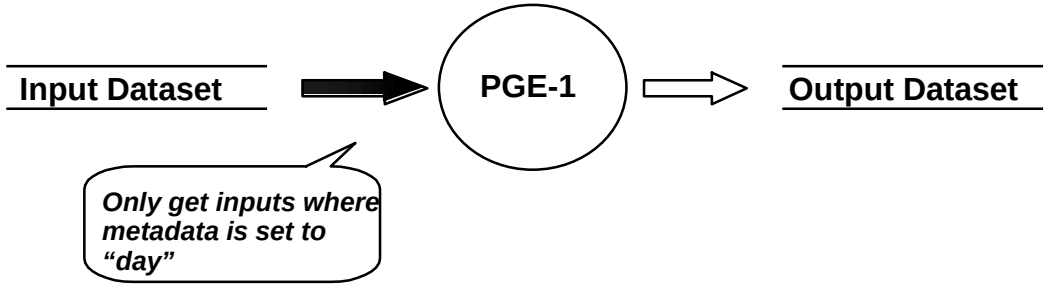


The following information needs to be filled out in the ESDT metadata file for each input with a metadata check:

- **Object = Metadata_Definition**
- **Parm_Name = <name of metadata parameter to check>**
- **Container_Name = <name of container or group that includes metadata parameter>**
- **Type = <type of parameter (int, float, string)>**
- **End Object = Metadata_Definition**

Metadata Query

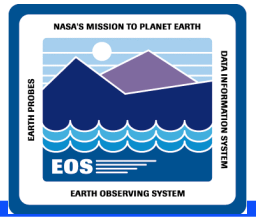
Use metadata values to further refine the list of inputs acquired for the PGE.



Production Rule Information Needed for PGE Registration:

- **Schedule Type and corresponding data.**
- **Metadata Query Objects for inputs to be "refined".**
- **Metadata Definition objects specifying the type of metadata parameter.**

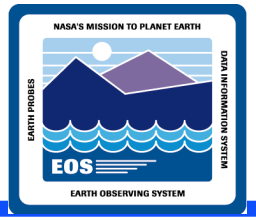
“Metadata Query” ODL Files and Parameters



The following information needs to be filled out in the PGE metadata file for each input with a metadata query:

- **Object = Metadata_Query**
- **Parm_Name = <name of metadata parameter to query against>**
- **Operator = <operator on parameter value>**
- **Value = <value to be queried against>**
- **Database_Query = <get value from PDPS database; valid values are: NONE, ORBIT NUMBER, PATH NUMBER, TILE ID, START DATA DAY END DATA DAY>**
- **End Object = Metadata_Query**

“Metadata Query” ODL Files and Parameters (CONT)



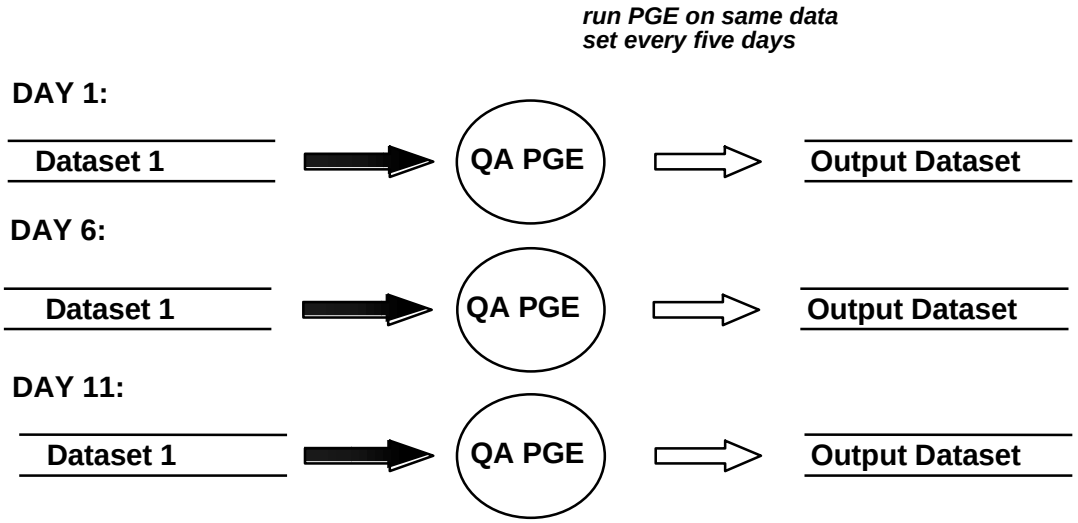
The following information needs to be filled out in the ESDT metadata file for each input with a metadata query:

- **Object = Metadata_Definition**
- **Parm_Name = <name of metadata parameter to query>**
- **Container_Name = <name of container or group that includes metadata parameter>**
- **Type = <type of parameter (int, float, string)>**
- **End Object = Metadata_Definition**

Intermittent Execution

Run a PGE every Nth time it is able to be run.

Note this is different than the case of running a monthly average once every month (which is covered by the basic temporal rule)

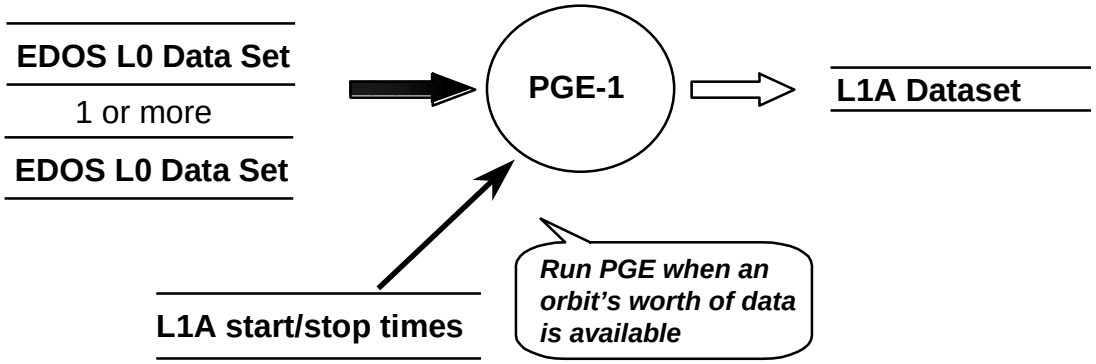


Production Rule Information Needed for PGE Registration:

- **None -- this is specified by choices on the Production Request Editor.**

Special Level 0 Processing Needs (Orbital Processing)

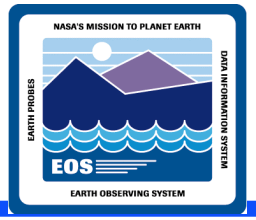
Identify and stage proper input Level 0 data to produce Instrument Team defined Level 1A granules.



Production Rule Information Needed for PGE Registration:

- **PGE Schedule Type is "Orbit Scheduled"**
- **Orbit Model ODL file.**
- **Path Model ODL file (if path mapping desired).**

“Orbit Scheduled” ODL Files and Parameters



The following information needs to be filled out in the PGE metadata file:

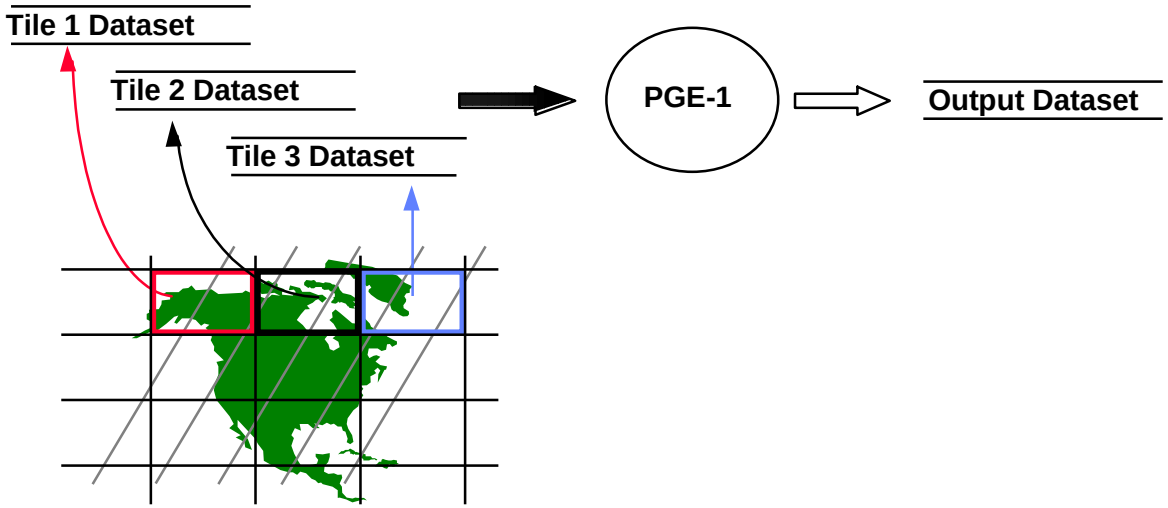
- **Schedule_Type = “Orbit”**
- **Processing_Boundary = “Start_Of_Orbit”**
- **Processing_Period = “Orbits=X” <where X is an integer value>**

The following information needs to be filled out in the ORBIT metadata file:

- **Platform = <platform of spacecraft>**
- **Orbit_Start = <start time of orbit>**
- **Orbit_Number = <number of orbit>**
- **Path_Number = <number of path (0-233) matching that orbit>**

Tiling

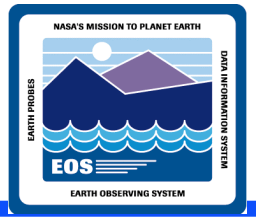
Identify and stage proper input data to produce Instrument Team defined tiles.



Production Rule Information Needed for PGE Registration:

- **PGE Schedule Type is “Tile Scheduled”**
- **Tile ODL file specified.**

“Tile Scheduled” ODL Files and Parameters



The following information needs to be filled out in the PGE metadata file:

- **Schedule_Type = “Tile”**
- **Tile_Scheme = <name of tile scheme defined in Tile ODL file>**

The following information needs to be filled out in the Tile metadata file:

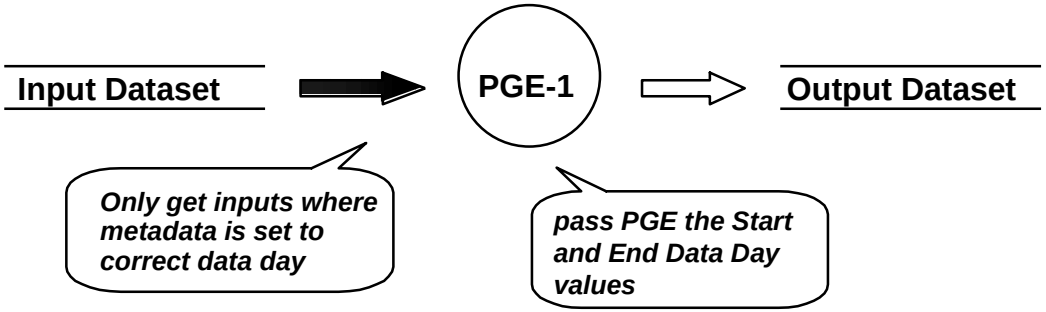
- **Tile_Scheme = <name of tile scheme>**

For each tile:

- **Object = Tile**
- **Tile_ID = <ID of tile>**
- **Tile_Description = <description of the tile>**
- **Coordinates = <the coordinates of the four (or more) corners of the tile>**

Data Day

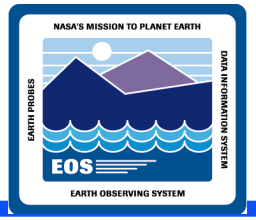
Acquire data for a given Data Day.



Production Rule Information Needed for PGE Registration:

- **PGE Schedule Type is "Time Scheduled"**
- **Create PCF Entry for Runtime parameters desired to have Start and End Data Day specified.**
- **Create Metadata Query object for inputs that are to be requested by Data Day.**

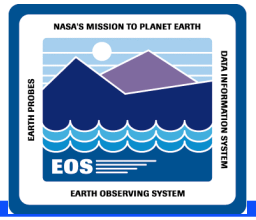
“Data Day” ODL Files and Parameters



The following information needs to be filled out in the PGE metadata file for each input that has to be retrieved by Data Day:

- **Object = Metadata_Query**
- **Parm_Name = <Data Day parameter name>**
- **Operator = “>=“**
- **Value = “0”**
- **Database_Query = “START DATA DAY”**
- **End Object = Metadata_Query**
- **Object = Metadata_Query**
- **Parm_Name = <Data Day parameter name>**
- **Operator = “<=“**
- **Value = “0”**
- **Database_Query = “END DATA DAY”**
- **End Object = Metadata_Query**

“Data Day” ODL Files and Parameters (CONT)



The following information needs to be filled out in the PGE metadata file. A PCF entry needs to be created for the runtime parameters that will specify Start and End Data Day values. The following parameter will allow PDPS to populate the runtime parameters with the Start and End Data Days:

Pge_Parameter_Dynamic_Value = “START DATA DAY”

or

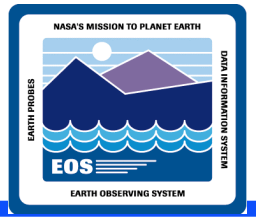
- **Pge_Parameter_Dynamic_Value = “END DATA DAY”**

“Data Day” ODL Files and Parameters (CONT)

The following information needs to be filled out in the ESDT metadata file for each input acquired by Data Day:

- **Object = Metadata_Definition**
- **Parm_Name = <name of Data Day parameter>**
- **Container_Name = “AdditionalAttributes” (because Data Day is in a PSA)**
- **Type = string**
- **End Object = Metadata_Definition**

Final Notes



- **Since the implementation of the Production Rules threads is not yet complete, we cannot guarantee that there will not be further refinements of the ODL templates.**

- **ODL templates can be found online at:**
- **http://dmserver.gsfc.nasa.gov/ecsdev/relb/pdps/index.html**
- **http://ecsinfo.hitc.com/iteams/iteams.html**

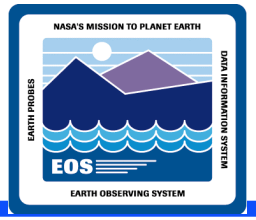
- **Detailed Production Rule Information**

Detailed Production Rule Information are covered in depth in publication: MODIS Science Data Processing Software Version 2.0 System Description, SDST-104, dated May 19, 1998.

Test Scenarios for selected Production Rules can be viewed by

accessing the SCF at:
</home/dheroux/DPS/TESTBED/MISR/SSIT/V2/ODL/Scenarios>

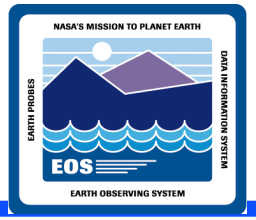
Science Software and Production Requests



Science software is one of the keys to production planning and processing:

- **Performs the actual data processing to create desired products.**
- **Is developed at Science Computing Facilities (SCFs) external to ECS.**
- **Is embodied in Product Generation Executives (PGEs) when the software is integrated into the ECS production processing environment.**
  - **PGEs are science software code (e.g., executable programs or shell scripts) that contain the instructions for processing data to create the desired products.**

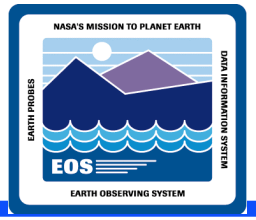
Science Software and Production Requests continued



The production request (PR) is another key to production planning and processing. The Production Planner defines ECS science data processing in terms of PRs.

- **A PR is an order for data to be produced by the Data Processing Subsystem.**
- **A single PR may specify several jobs (using the same PGE) that are to be run over a period of time or a single job producing a single set of data.**
- **PRs may apply to the processing of new data (standard PRs or standing orders) or the reprocessing of existing data (reprocessing PRs).**

Science Software and Production Requests continued



- **Each PR identifies a specific PGE for generating a particular type of product.**
  - **Some PGEs are dependent on others; i.e., some PGEs require input data that are the output of other PGEs.**
  - **The planning software will recognize and reject a PR when the PR specifies a PGE that requires data from another PGE that has not yet been specified in a PR.**

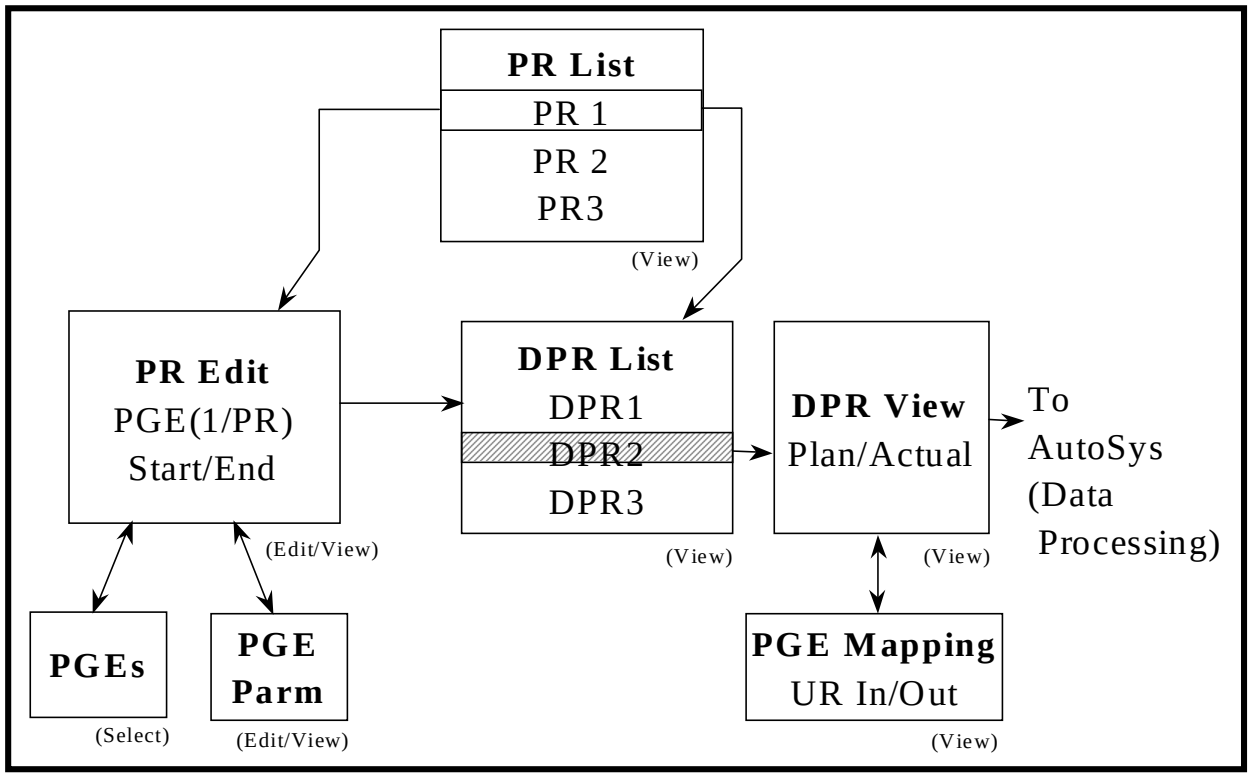
Science Software and Production Requests continued

The Planning Subsystem performs the following functions:

- Uses each PR to generate either one or a series of Data Processing Requests (DPRs).
  - Each DPR corresponds to one execution of a single PGE.
  - Each DPR contains the information that is needed by the SDP processing function, including PGE-related information.
- Checks the availability of the data required for the DPR, either from the data server (if the data have been previously ingested) or from internal predictions (if the data are expected to arrive in the future).
- Determines what data will be included in the DPR output so the system can make predictions concerning the availability of data for subsequent PGEs.

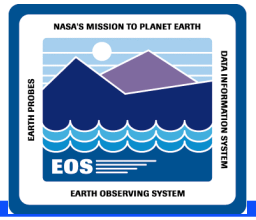
The figure on next slide shows the relationships among the PGEs, PRs, and DPRs as they are accessed through the Production Request Editor GUI.

Science Software and Production Requests continued



Production Request Editor Flow

Science Software and Production Requests continued



Types of Processing

ECS either accommodates or will accommodate the following three general types of data processing:

- **Routine Processing**
- **Reprocessing**
- **Ad-Hoc Reprocessing**
- **On-Demand Processing**

Routine processing is pre-defined software production processing that is periodic and keyed to data arrival. For example, every day a Production Planner includes in the daily schedule a DPR for generating a particular Level 1A product from the most recent Level 0 data from the applicable satellite instrument

Science Software and Production Requests continued

Reprocessing typically involves using a new, improved PGE to process data that had previously been processed with an older version of the PGE. In such cases reprocessing would be a large-scale operation, especially if several years worth of data were to be reprocessed. Consequently, the Production Planner is likely to schedule reprocessing in manageable quantities so the processing resources can accommodate routine and on-demand processing in addition to the reprocessing.

In addition, ad-hoc reprocessing could be necessary at any time. For example, if a product fails a quality assurance (QA) check, the same PGE could be run again on the same data set in the hope of creating an acceptable product. Similarly, if processing of a PGE fails for some reason, it might be possible to rerun the PGE and hopefully achieve a successful outcome.

Science Software and Production Requests continued

On-demand processing is ad-hoc processing initiated by either the Planning Subsystem or an end-user (as opposed to the Production Planner). For example, a researcher using data from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument on the Terra satellite may need a particular Level 2 product that has not yet been generated. The ASTER researcher would submit an on-demand request to have the product generated from a Level 1B product stored in the archive.

In the future such on-demand processing requests (OPRs) will be entered from a Client Subsystem tool, passed through the Distributed Information Manager (Data Management Subsystem) and the Data Server to the Planning Subsystem. Currently there is a work-around to the automated process which requires the requester to contact DAAC personnel to make the request. So far ASTER researchers are the only identified external users of automated on-demand processing.

DPREP

Introduction

This section contains information to run DPREP.

- **DPREP is made up of three PGE's each run separately.**
- **The PGE's are titled Step1 DPREP, Step 2 DPREP and Step 3 DPREP.**
- **The input files normally come from INGEST.**
- **These files are depicted in the three step DPREP process in the diagram depicting the DPREP Processes.**
- **The output files generated from each of the DPREP PGE's contains Ancillary Attitude, and Ephemeris data that becomes new inputs to Instrument PGE's.**

DPREP Introduction continued

- **These Instrument PGE's will then process its satellite data with similar time span files created by DPREP.**
- **The DPREP registration process for each of the three PGE's creates in the Science Data Server Archive a subscription for each of the DPREP PGE's.**
- **PGE execution then takes place in the PDPS.**
- **The SSI&T effort for DPREP PGE's is similar in effort to what would be required to register any other PGE.**

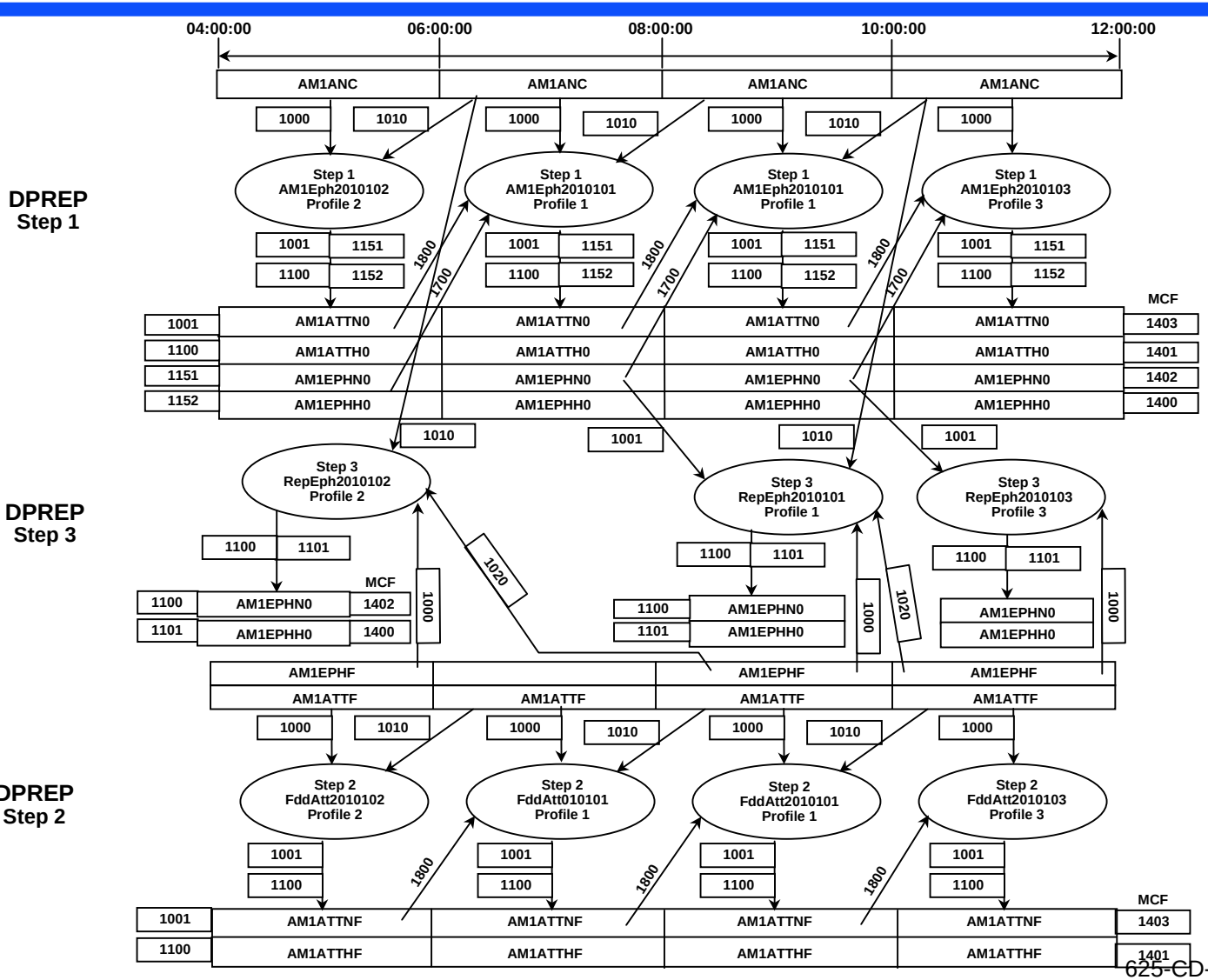
SSI&T Activity for DPREP

- **SSI&T Activity for DPREP**
- **The Level Zero datasets are received in 2 hour chunks.**
- **The file processes diagram depicts the minimal time span allowable for a DPREP run. In a normal operation of DPREP, a twenty four hour time span would be prepared for. This would require additional 2 hour chunks and thus additional files of data would need to be registered.**
- **Before the registration process can take place, a number of files will have to be updated to process a block of data for a particular time period. Therefore, DPREP input files will have to be identified and various templates for the SSI&T process will require annotation.**

SSI&T Activity for DPREP Cont

- **The sections that follow in the Training Manual have been highlighted with notations as to what SSI&T process applies in the preparation of each template and the function required to register each section.**
- **With a particular function identified, other portions of this manual can be referred to for more detailed precedures to be used to carry out the full SSI&T process.**
- **Whenever new input files are introduced or updated executables are re-introduced it is wise to first set up the PGE to run from the Command Line. This will determine if what has been introduced will run error free.**
- **After a successful Command Line run it is advisable then to complete the SSI&T effort to run from the PDPS. Command Line Runs include the use the PCF to run from in the Science Data Server. PDPS runs include ESDT's and ODL files to generate internal PCF's.**

DPREP File Processes



DPREP Processes

DPREP Processes and Procedures

Processes and procedures are provided in the following files on an SGI machine used to support SSI&T:

- **DPREP README, HowToCreateDprepTarFile and HowToRunDPREP files located at :/usr/ecs/TS1/CUSTOM/data/DPS/**
- **DPREP binary located: :/usr/ecs/TS1/CUSTOM/bin/DPS/**

DPREP consists of three pges each run separately.

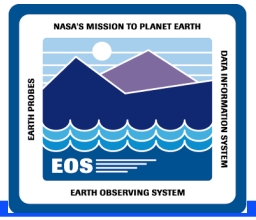
1 The first step is a ksh script called EcDpPrAm1EdosEphAttDPREP_PGE, which serves as a driver for three executables:

- **EcDpPrAm1EdosAncillary**
- **EcDpPrAm1EdosEphemerisRepair**
- **EcDpPrAm1ToolkitToHdf**

2 The second step is EcDpPrAm1FddAttitudeDPREP_PGE.

3 The third step is EcDpPrAm1FddEphemerisDPREP_PGE.

Updating the PDPS Database and Data Server



Integration of Science Software with ECS requires that information about PGEs be made known to the PDPS in its database.

- **PDPS needs information to plan, schedule, and run science software.**

- **PDPS Database and Data ServerTools are accessible from SSIT Manager GUI.**

  Tools → PDPS Database → PCF ODL Template
  → SSIT Science Metadata Update
  → SSIT Opnl Metadata Update

  Tools → Data Server → Register Subscription
  → Insert Static
  → Insert Test Dynamic
  → Insert EXE TAR

Updating the Orbit Model

Introduction

To determine realtime the latest:

Orbit Start times, Orbit Period, Path Number and Orbit Number,

- **PDPS takes in specific information about the orbit of the satellite during initial SSI&T. This information then becomes the basis for predictions of future orbit start times and numbers.**
- **Because this value is accurate within a fraction of a second of time, the satellite may "drift" or a correction to orbit, known as a "burn" may have been applied. Therefore, the satellite Orbit Start Time can get out of sync either +/- with reality.**
- **The consequences are an elapse in time that will affect the Production Request Editor's ability to find a granule that should match with a DPR, or an incorrect Orbit Time could be passed to the PGE.**
- **The update of Orbit parameters will be done weekly at a specific time with scrips specifically written to extract the new Orbit Parameters from the most recent DPREP output file.**

Updating the Orbit Model Continued

- These parameters intern will be inserted manually to the ORBIT.ODL file and then the re-registration of the Orbit.ODL file into the PDPS by SSI&T personnel.

- The M&O support Help Desk Team is responsible for knowing when changes to Orbit location have taken place from the Flight Dynamics Systems (FDS).

- A KnowledgeBase with backup procedures will be maintained by M&O for contingencies concerning Orbit Model updates.

- DPREP processing will be the most likely place to experience a failure due to Orbit time sync error encounters. The restoration of Orbit parameters with new values from FDS will most likely be necessary. The following procedures are provided to bring about an updated Orbit Model within ECS.

Updating the Orbit Model Continued

Procedures to Update the Orbit Model

Upon receipt of updated orbit parameters: ORBIT_ NUMBER, ORBIT_PERIOD, ORBIT Path Number and ORBIT_START Time.

Proceed with the following steps.

---

1 **Telnet or Rlogin to location (ais) system where ODL files are stored. ie; "/usr/ecs/OPS/CUSTOM/data/DPS/ODL"**

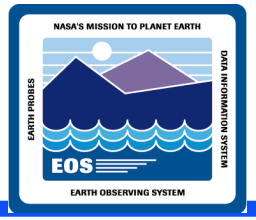
2 **Select the ORBIT.odl that is currently being used.**

3 **Using vi, update the following files with the new parameter values received:**

- **ORBIT_AM1.odl and/or ORBIT_EOSAM1.odl if they both are in use.**

4 **Have someone double check your entries for accuracy before preoceeding to the SSIT Manager for registering the new ODL file in the PDPS system.**

Updating the Orbit Model Continued



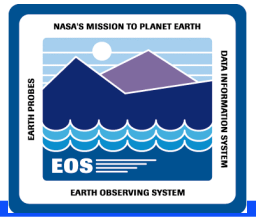
5 For Test Data only; determine the Instrument PGE ODL that will be updated. MISR, MODIS etc.

- **Using vi, update the corresponding PATHMAP_Instrument_.odl file with the new parameter values received.**
- **Ensure that the ABSOLUTE_PATH and MAPPED_PATH parameters agree with those in the new ORBIT_XXXX.odl.**

6 SSI&T personnel will execute an Orbit Model Update by running a Dummy PGE established for this purpose at each of the DAAC's. Note: A dummy PGE is ran since a normal PGE cannot be re-registered if any DPRs exist in the system.

Notify DAAC Operations Supervisor that the Orbit Model has been updated. He will make a log entry of such action taken and may request the old computed values and the new replacement values be provided. The Supervisor will ensure that the orbital change is within several seconds, the expected change and not minutes

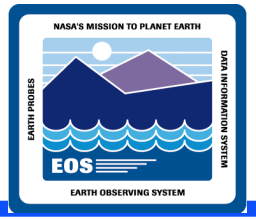
Updating the PDPS Database with ESDT Metadata



PDPS needs basic information on every type of file associated with PGEs.

- **Metadata for the PDPS Database is first prepared in Object Definition Language (ODL), one ODL file for each ESDT.**
- **Determine Science Data Server ShortName for ESDT corresponding to file.**
- **Search ESDT directory for ESDT ODL file. If file exists, there is no need to make another one for this ESDT. The file naming convention is ESDT_*ShortName*.odl, where *ShortName* is the same name used for the Science Data Server.**
- **If not, copy the ESDT ODL template from configured area to user space.**
- **Add required metadata to ODL file via text editor.**
- **ShortName in ODL file must match ShortName of file itself, ShortName in ESDT descriptor, and ShortName in PGE metadata ODL file.**
- **Copy the ESDT ODL file to the ECS configured area.**

Updating the PDPS Database with PGE Metadata



PDPS needs basic information or metadata on the PGE.

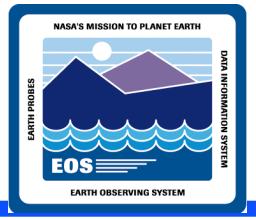
Order for this Update PGE Metadata Activity: It is recommended that all ESDT metadata ODL files associated with the PGE be prepared and put into the configured area before this procedure is executed.

Invoke PCF ODL Template Tool.

This tool prompts the user for the following information:

- **Configuration file - use default ConfigFile path and filename for DAAC**
- **Process Control File - Path and filename of PCF, default path is current**
- **PGE Name - Name of PGE associated with PCF**
- **PGE Version - PGEversion, default needs to be determined if not known**

Updating the PDPS Database with PGE Metadata (cont.)



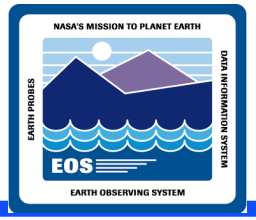
Many PCF ODL files can be made on same invocation until user quits.

Program outputs a file with name PGE_*PGEname#PGEversion*.tpl.

Go to full path from which SSIT Manager is run and change this output ".tpl" filename extension to ".odl" or copy template into file with same name and ".odl".

- **Edit PGE_*PGEname#PGEversion*.odl file to add metadata.**

Updating the PDPS Database with PGE Metadata (cont.)



Process Steps:

- **From SSIT Manager select Tools → PDPS Database → SSIT Science Metadata Update.**

The program prompts for the following information:

- **ConfigFile - use default for path and filename at each DAAC.**
- **mode - use default of ops.**
- **PGE name - name of PGE that will be registered (user choice).**
- **PGE version - PGE version to be registered.**
- **Quit out of program.**

For Version 2 the mode corresponds to ops or TS1 PDPS Database.

Updating the PDPS Database with Operational Metadata

Assumption:

All ESDT metadata ODL files associated with the PGE must already be prepared and put into the configured area and the PGE must be registered using the Science Metadata Update Tool.

Process Steps:

- **Invoke the PDPS/SSIT Database Update GUI Tool.**
- **Select the PGE name and version.**
- **Select new PGE and Done. PDPS needs basic operational metadata on the PGE to plan resources.**
- **Select Profile and enter values in fields under Performance Statistics:**

Wall clock time	**CPU time**
Max memory used	**Block input ops**
Block output ops	**Swaps**
Page faults	

Updating PDPS with Operational Metadata

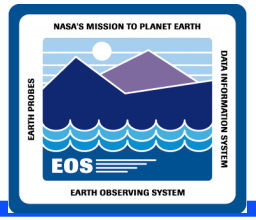
Process Steps (cont.):

- **Select Resource Requirements and enter values: Max disk space used during PGE run.**
- **Select Proc. String. Only one should be listed. Number of Processors should be 1.**
- **Select Apply to update the PDPS database.**
- **To start over, use RESET button.**

The performance statistics collected by running the PGE under EcDpPrRusage will be used to initialize the PDPS database so that the Planning for DPRs can be performed.

The actual values entered can be approximate.

Placing Dynamic Data Granules on the Science Data Server



- A granule of data is the smallest aggregation of data that is individually managed and archived in the ECS. When products are requested through PDPS, the PGE will be run using the required input data granules acquired from the Science Data Server.
- Dynamic test data granules are delivered with the PGE for input at runtime.
- Insert of dynamic test data is done by a Science Data Server Program.
- DAPs contain MCF templates or samples for output products, not input.
- PGEs, through the SDP Toolkit, generate target MCFs for each data granule produced using the source MCF ODL files delivered with the PGE.
- The target MCFs produced by the PGE are used by the Data Server to insert data products, thus they are called database load ODL files.
- SSI&T will have to create target MCFs (database load ODL files) for input test data granules since they are not produced by a PGE.

Creating a Target MCF for a Dynamic Data Granule

Assumption: All ESDT metadata ODL files associated with the PGE must already be prepared and put into the configured area and the PGE must be registered using the SSIT Science Metadata Update Tool.

Process Steps to place dynamic data granules on the Science Data Server:

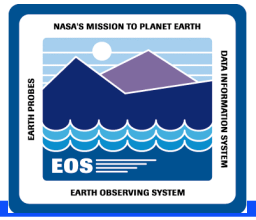
- **If a source MCF template is not available for the dynamic data granule, then make one by editing a template provided for SSI&T.**
- **Creating a target MCF for dynamic data granule from the source MCF.**
  - **Go to the directory where the source MCF resides.**
  - **Invoke SrcToTargetMCF program from command line.**
  - **Enter source MCF filename (.mcf) and target MCF filename (.met).**
  - **Edit TargetMCFfilename.met file.**
- **For all Data_Location="PGE" attribute in Source MCF, enter data values.**
- **Date values are provided by Instrument Teams on delivery of the DAP.**
  - **Data granule start date/time and end date/time must be provided.**
- **Save the file.**

Inserting Dynamic Data Granules to the Science Data Server

Process Steps for inserting Dynamic Data Granules into the Science Data Server:

- **Go to the directory where the dynamic data granule resides.**
- **From the SSIT Manager GUI →Tools → Data Server → Insert Test Dynamic.**
- **The Insert Test Dynamic program will be running.**
- **Enter the following information:**
  - **ConfigFileName - use default.**
  - **ESDT ShortName - ESDT ShortName corresponding to data granule.**
  - **Filename to Insert - Full path & filename of data granule to be inserted.**
  - **Associated ASCII metadata (target MCF) filename to Insert - same as above filename but with ".met" extension.**

Placing Static Data Granules on the Science Data Server



Static data granules are those whose temporal locality is static over long periods of time. Examples are calibration files which change only with a new version of the PGE.

Static test data granules are delivered with some PGEs.

Source MCFs delivered with PGEs are treated as static data granules.

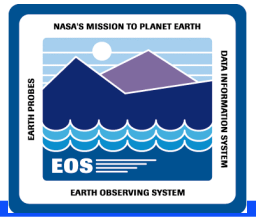
- **Source MCFs delivered with PGEs can be used from the command line to test a PGE, however an MCF will have to be generated from the SDSRV for use in full PGE integration.**

Insert of static test data is done by a Data Server Program in SSIT Manager.

DAPs contain MCF templates only for output products, not for input files.

PGEs, through the SDP Toolkit, generate target MCFs for each data granule produced using the MCF files generated by the SDSRV.

Placing Static Data Granules on the Science Data Server (cont.)



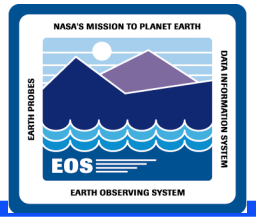
Target MCFs (database load ODL files) are needed to insert all data files, including static data granules, to the Science Data Server.

SSI&T will have to create target MCFs (database load ODL files) for static data granules delivered for SSI&T since they are not produced by a PGE.

A template for static Metadata ODL files is provided for SSI&T.

The static version has parameters unlike those for dynamic data granules.

Creating a Metadata ODL File for a Static Data Granule



Assumption: All ESDTs associated with the PGE, including either a single ESDT for static type files or an ESDT for each static file, must already be registered. PGE must be registered using Science Metadata Update Tool.

A template metadata ODL file which can be edited is provided for SSI&T by the instrument team.

Process Steps to place static data granules on the Data Server:

- **Create a metadata ODL file for a static data granule.**
  - **Go to the directory where the metadata ODL template resides.**
  - **Copy StaticODLmet.tpl to the user work directory as *filename*.met.**
  - **Edit the *filename*.met and enter the following information and save:**

    ShortName - ESDT ShortName

    VersionID - ESDT VersionID

    ParameterName - name of static file in InformationContentContainer

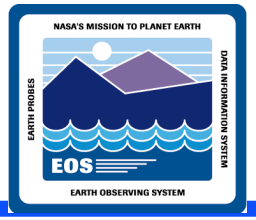
    ParameterValue - Cn: C = coefficient file, M = MCF; n = 1,2,...

Inserting Static Data Granules to the Data Server

Process Steps for inserting Static Data Granules to the Data Server:

- **Go to the directory where the dynamic data granule resides.**
- **From the SSIT Manager GUI →Tools → Data Server → Insert Static**
  - **The Insert Static program will be running.**
- **Enter the following information:**
  - **Config Filename - use default.**
  - **Mode - use default mode “ops”**
  - **ESDT ShortName - ESDT ShortName for data (bucket ESDT).**
  - **Science Group - Cn: C = coefficient file, M = MCF; n = 1,2,…**
  - **PGE Name - Name of PGE registered.**
  - **PGEVersion - PGE version or take default of 1.**
  - **Filename to Insert - Filename of static data granule to be inserted.**
  - **Associated ASCII metadata (target MCF) filename to Insert - same as the above filename but with “.met” extension.**
- **The static granule will be inserted. The program can run again until the user quits.**

Inserting Science Software Executable on the Science Data Server



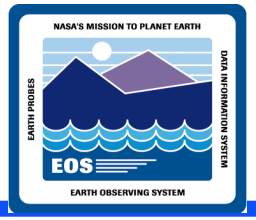
All science software executables must be inserted on the Data Server to be run by the PDPS. The executable package is called a SSEP or EXE Tar. The steps of this procedure involve three activities:

Assembling a Science Software Executable Package:

- **Make a new directory to hold the contents of the SSEP.**
- **Copy all files to go into the SSEP into this directory: PGE executables shell scripts, SDP Toolkit message files.**
- **Use UNIX tar to make the package**

- **Copy over the Target MCF template to *filename*.met, edit and save:**
  - **The PGE name is PGEEXE. Enter PGE version and parameter values according to program prompt.**
- **Select from the SSIT Manager GUI: Tools → Data Server → Insert EXE TAR.**
  - **Enter PGE Name, SSWVersion, SSEPFileName, ExecFileName**

PGE Planning and Processing

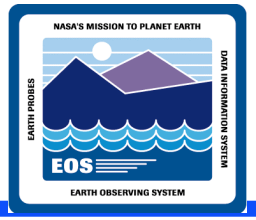


After the PGE has been linked to the DAAC Version of the SDP Toolkit, all associated ESDT and PGE information has been entered into the PDPS Database, all operational metadata has been entered, and the PGE has been registered, then the PGE is ready to be run in PDPS under AutoSys.

The major steps in the PGE Planning and Processing are the following:

- **Register a subscription for test input and output files (once per ESDT).**
- **Using the Production Request Editor, enter processing information and submit a Production Request.**
- **Using the Planning Workbench, create a new production plan, schedule the processing, activate the plan, and review the planning timeline.**
- **Monitor production under AutoSys.**
- **Using the QA monitor, acquire and view the test output file from the Data Server and examine the Production History File.**

Registering Subscriptions for Test Input and Output Files

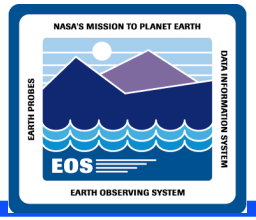


The Data Subscription Management application is used to manage the receipt of a subscription notification from the Data Server.

Process Steps:

- **Invoke the Subscription Editor from the SSIT Manager.**
- **From the SSIT Manager GUI →**
  - **Tools → Data Server → Register Subscription**
- **Register a subscription for each of the input files and output product files associated with the PGE.**

Using the Production Request Editor

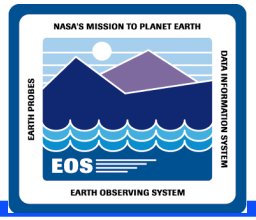


The Production Request Editor is a GUI tool which provides the capabilities of submitting a Production Request (PR), looking at production requests in the system, and viewing the Data Production Requests (DPRs) which have been expanded from the original PR.

It is invoked from the command line as illustrated in two sections of the accompanying Training Manual 625, titled: Creating a New Production Request and Using the Production Request Editor.

Only one PR can be submitted at a time by saving a PR file which is then known to the Planning, Scheduling and Production System. When a PR is submitted, the windows are re-initialized and another PR can be submitted.

The Production Request Editor GUI



TAB DESCRIPTIONS

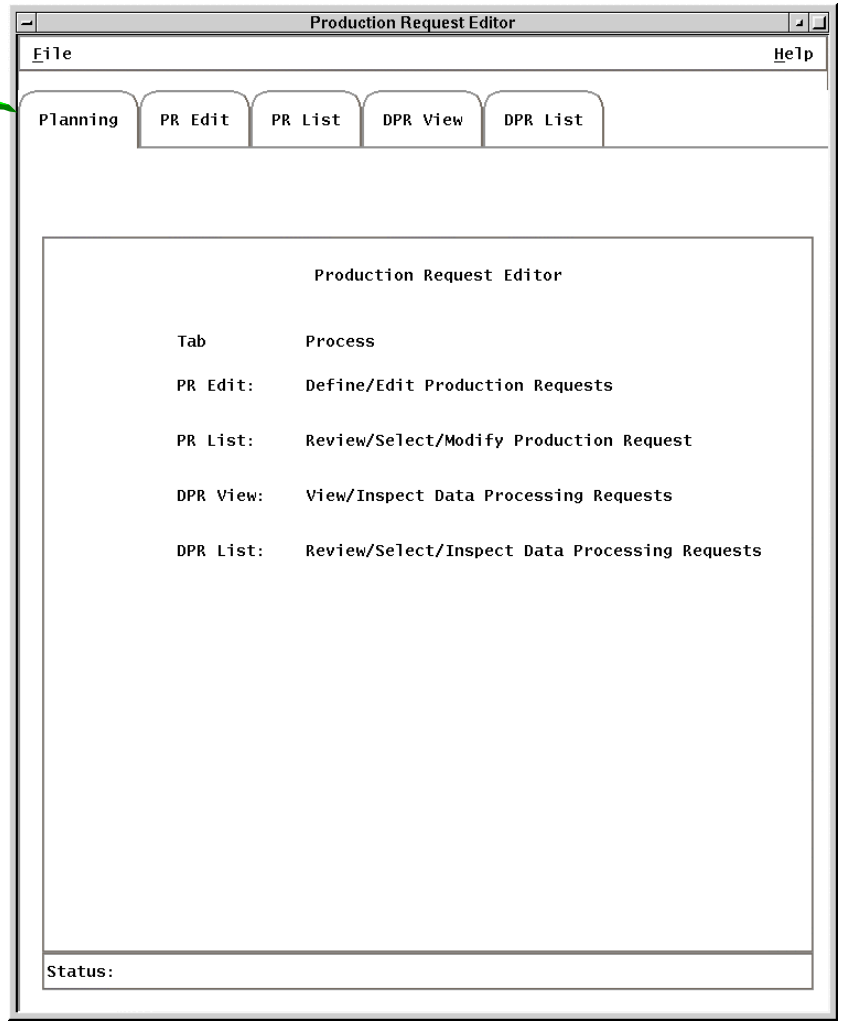
Planning - Displays a list of the four capabilities provided by PR Editor

PR Edit - Define and edit Production Requests.

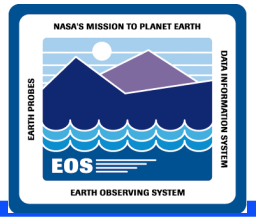
PR List - Displays a list of all PRs entered into the system.

DPR View - Displays detailed information for a selected DPR.

DPR List - Displays all DPRs associated with a selected PR.



Using the PR Edit GUI

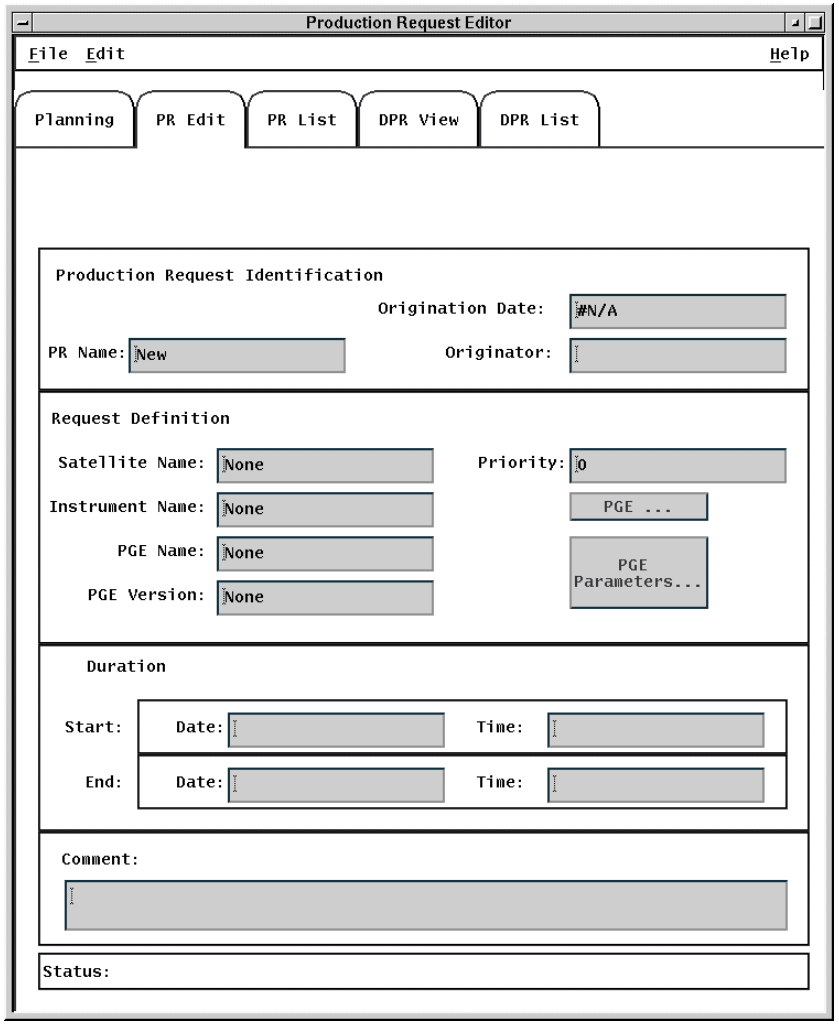


Selecting the PR Edit tab at the top of the main Production Editor GUI.

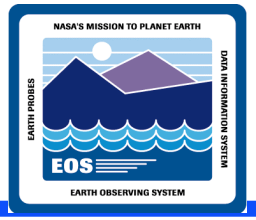
The following information must be entered on the PR Edit screen:

- **PR Name - enter as New when PR is being done.**
- **Satellite Name - Name of spacecraft.**
- **Instrument Name- Name of Instrument for which data is being processed.**
- **PGE Name - Name of the PGE registered in the system.**
- **PGE Version - Version of the PGE corresponding to PGE name.**
- **Originator - Name on Instrument Team or data provider.**
- **Priority - Priority to be assigned to this production request.**
- **StartDate - Start date for data coverage.**
- **StartTime - Start time for data coverage.**
- **EndDate - End date for data coverage.**
- **EndTime - End time for data coverage.**

PR Edit GUI


Production Request Editor
File Edit
Help
Planning
PR Edit
PR List
DPR View
DPR List
Production Request Identification
Origination Date:
#N/A
PR Name:
New
Originator:
Request Definition
Satellite Name:
None
Priority:
0
Instrument Name:
None
PGE ...
PGE Name:
None
PGE Version:
None
PGE Parameters...
Duration
Start:
Date:
Time:
End:
Date:
Time:
Comment:
Status:

Using the PR Edit GUI (2)



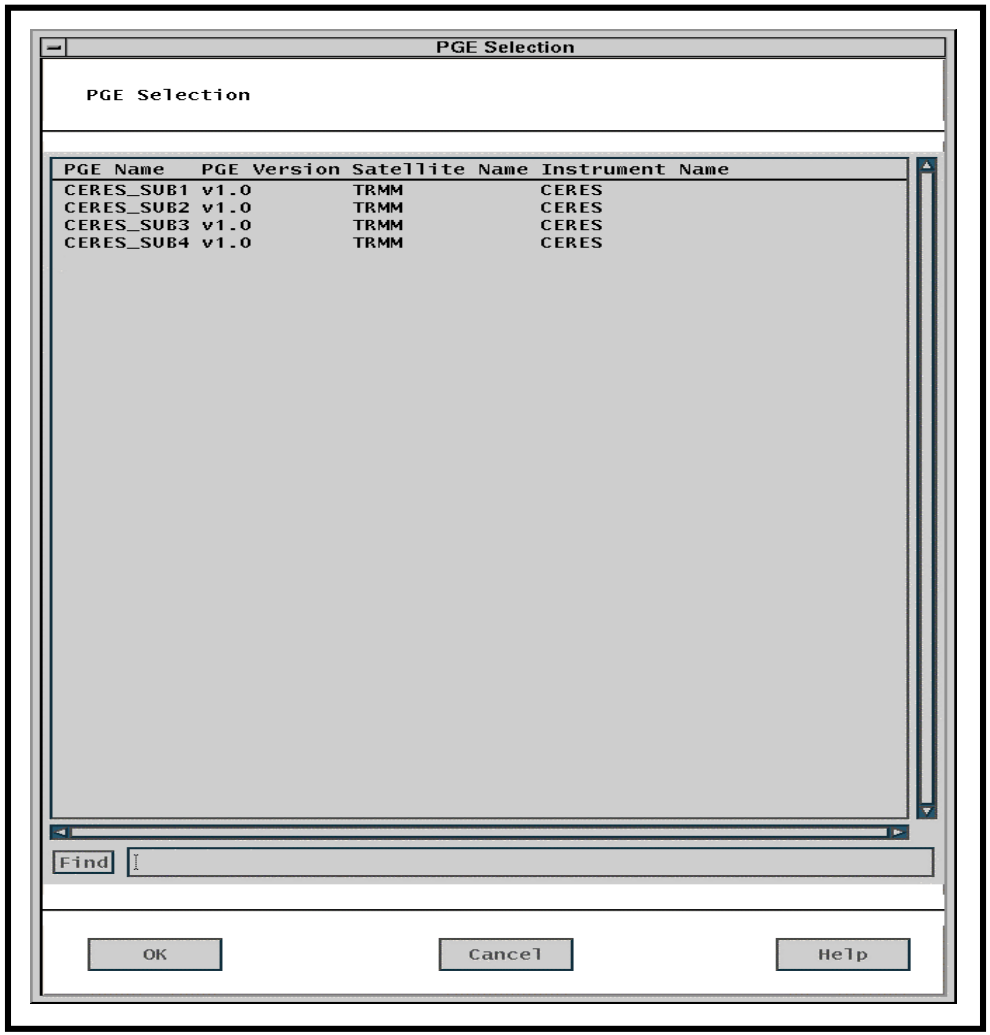
The PGE information is not entered directly into the PR Edit screen. To enter the PGE information, a pull down GUI is available by selecting "PGE…". Selecting a PGE from this GUI, enters the related information into the PR Edit screen.

If PGE Parameters are to be examined or changed, a pull down GUI is available by selecting "PGE Parameters…". The new value may be entered in the override box in this GUI.

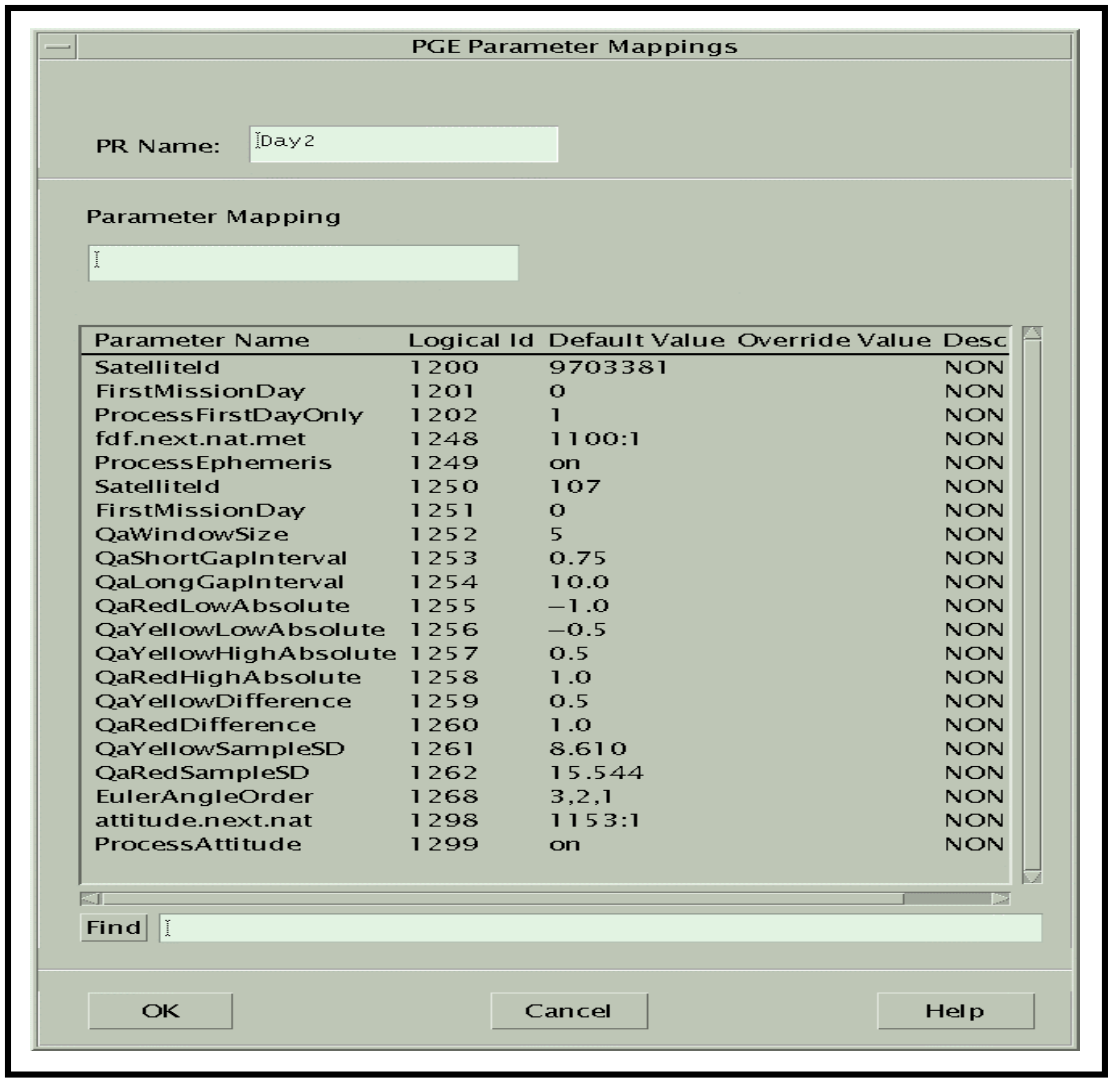
Entry of a comment is optional.

When the PR is complete, the "File" tab at the top is selected and a PR filename is entered in the "Save As" box.

PGE Selection GUI



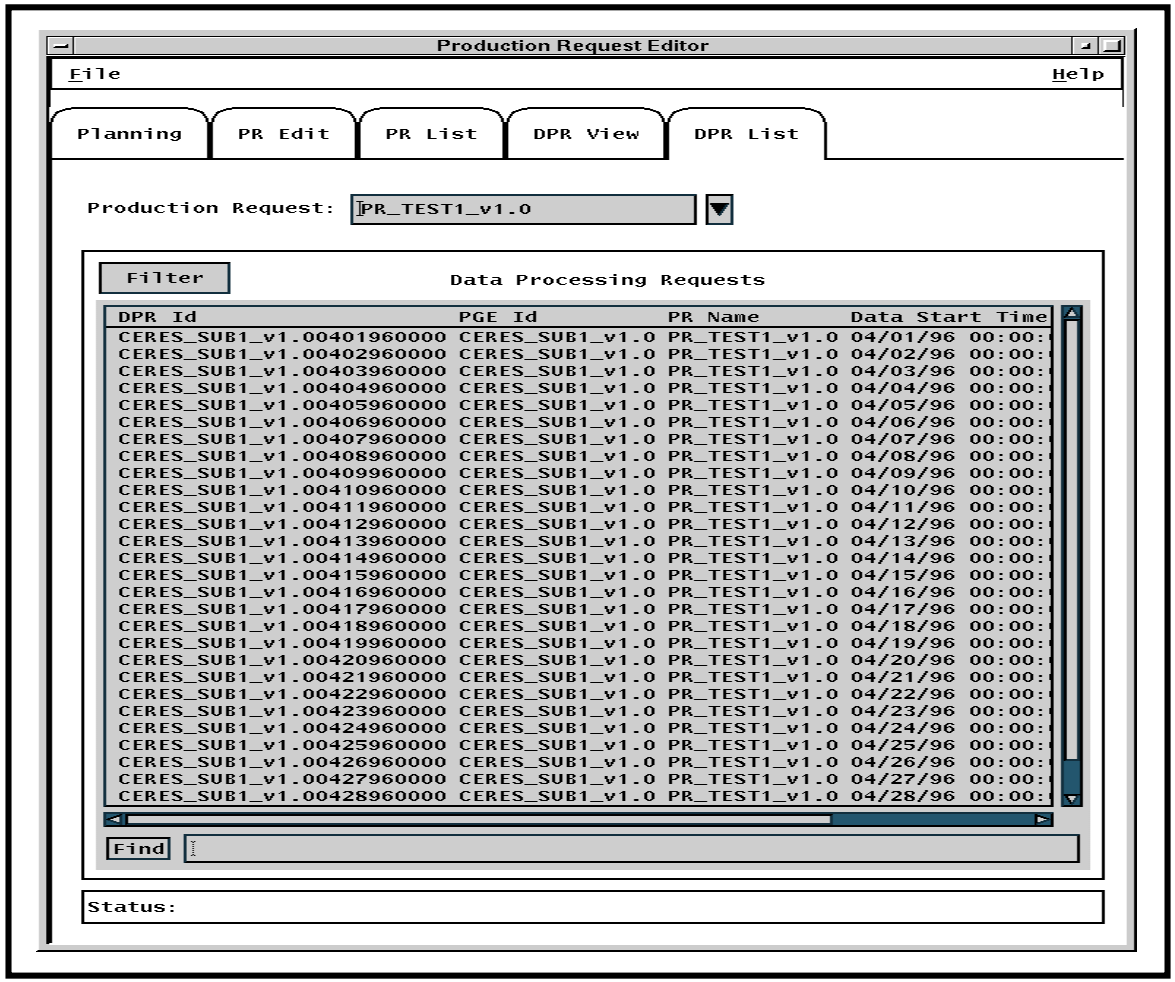
PGE Parameter Mapping GUI



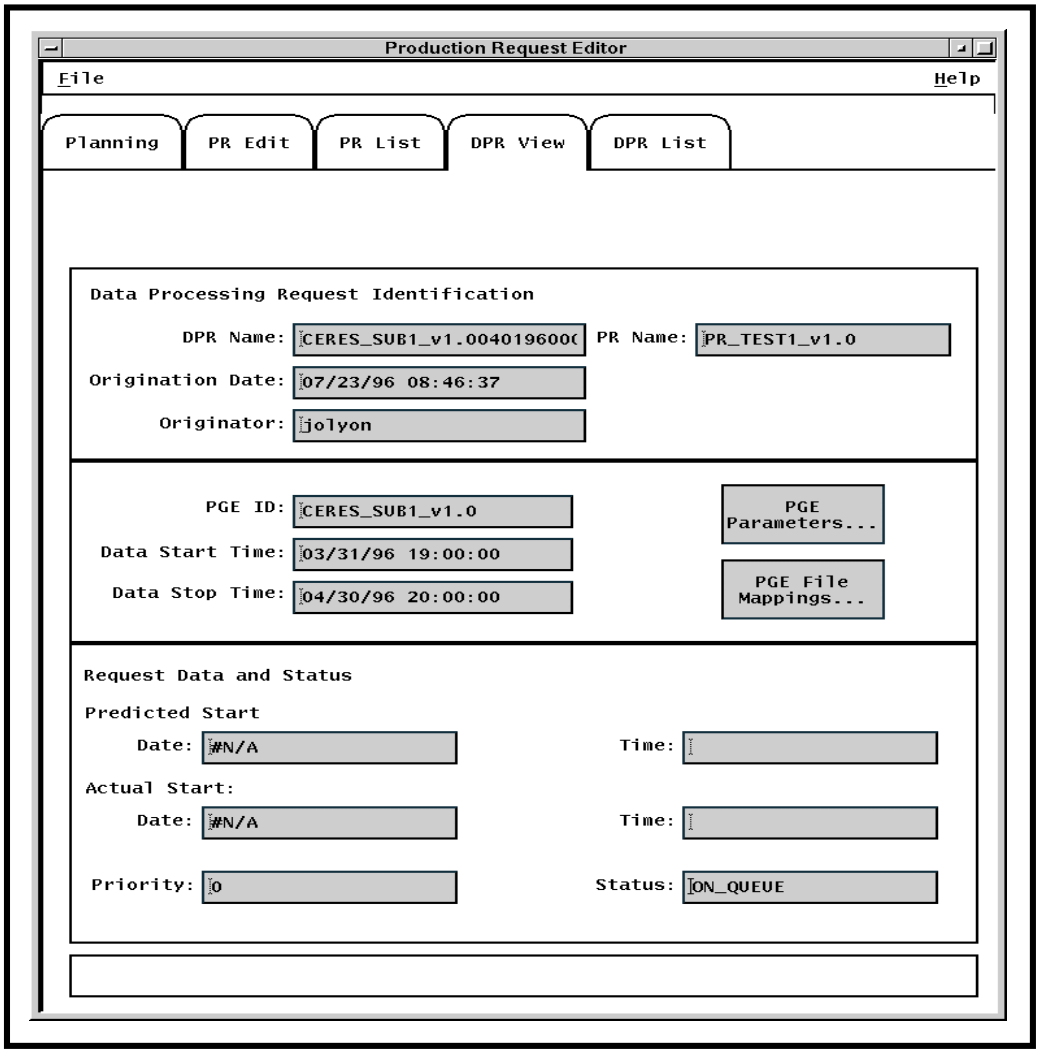
PR List GUI


Production Request Editor
File Edit
Help
Planning
PR Edit
PR List
DPR View
DPR List
Production Requests
PR Name PGE ID Priority Start End
PR_CERESS1_v1.0 CERES_SUB1_v1.0 1 03/21/96 19:00:00 03/22/96 19:
PR_TEST1_v1.0 CERES_SUB1_v1.0 1 03/31/96 19:00:00 04/30/96 20:
PR_TEST2_v1.0 CERES_SUB2_v1.0 1 03/31/96 19:00:00 04/30/96 20:
PR_TEST3_v1.0 CERES_SUB3_v1.0 1 03/31/96 19:00:00 04/30/96 20:
PR_TEST4_v1.0 CERES_SUB4_v1.0 1 03/31/96 19:00:00 04/30/96 20:
Find
Status:

DPR List GUI


Production Request Editor
File
Help
Planning
PR Edit
PR List
DPR View
DPR List
Production Request:
PR_TEST1_v1.0
Filter
Data Processing Requests
DPR Id
PGE Id
PR Name
Data Start Time
CERES_SUB1_v1.00401960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/01/96 00:00:
CERES_SUB1_v1.00402960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/02/96 00:00:
CERES_SUB1_v1.00403960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/03/96 00:00:
CERES_SUB1_v1.00404960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/04/96 00:00:
CERES_SUB1_v1.00405960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/05/96 00:00:
CERES_SUB1_v1.00406960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/06/96 00:00:
CERES_SUB1_v1.00407960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/07/96 00:00:
CERES_SUB1_v1.00408960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/08/96 00:00:
CERES_SUB1_v1.00409960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/09/96 00:00:
CERES_SUB1_v1.00410960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/10/96 00:00:
CERES_SUB1_v1.00411960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/11/96 00:00:
CERES_SUB1_v1.00412960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/12/96 00:00:
CERES_SUB1_v1.00413960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/13/96 00:00:
CERES_SUB1_v1.00414960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/14/96 00:00:
CERES_SUB1_v1.00415960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/15/96 00:00:
CERES_SUB1_v1.00416960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/16/96 00:00:
CERES_SUB1_v1.00417960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/17/96 00:00:
CERES_SUB1_v1.00418960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/18/96 00:00:
CERES_SUB1_v1.00419960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/19/96 00:00:
CERES_SUB1_v1.00420960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/20/96 00:00:
CERES_SUB1_v1.00421960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/21/96 00:00:
CERES_SUB1_v1.00422960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/22/96 00:00:
CERES_SUB1_v1.00423960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/23/96 00:00:
CERES_SUB1_v1.00424960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/24/96 00:00:
CERES_SUB1_v1.00425960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/25/96 00:00:
CERES_SUB1_v1.00426960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/26/96 00:00:
CERES_SUB1_v1.00427960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/27/96 00:00:
CERES_SUB1_v1.00428960000 CERES_SUB1_v1.0 PR_TEST1_v1.0 04/28/96 00:00:
Find
Status:

DPR View GUI


Production Request Editor
File
Help
Planning
PR Edit
PR List
DPR View
DPR List
Data Processing Request Identification
DPR Name:
CERES_SUB1_v1.004019600(
PR Name:
PR_TEST1_v1.0
Origination Date:
07/23/96 08:46:37
Originator:
jolyon
PGE ID:
CERES_SUB1_v1.0
Data Start Time:
03/31/96 19:00:00
Data Stop Time:
04/30/96 20:00:00
PGE Parameters...
PGE File Mappings...
Request Data and Status
Predicted Start
Date:
#N/A
Time:
Actual Start:
Date:
#N/A
Time:
Priority:
0
Status:
ON_QUEUE

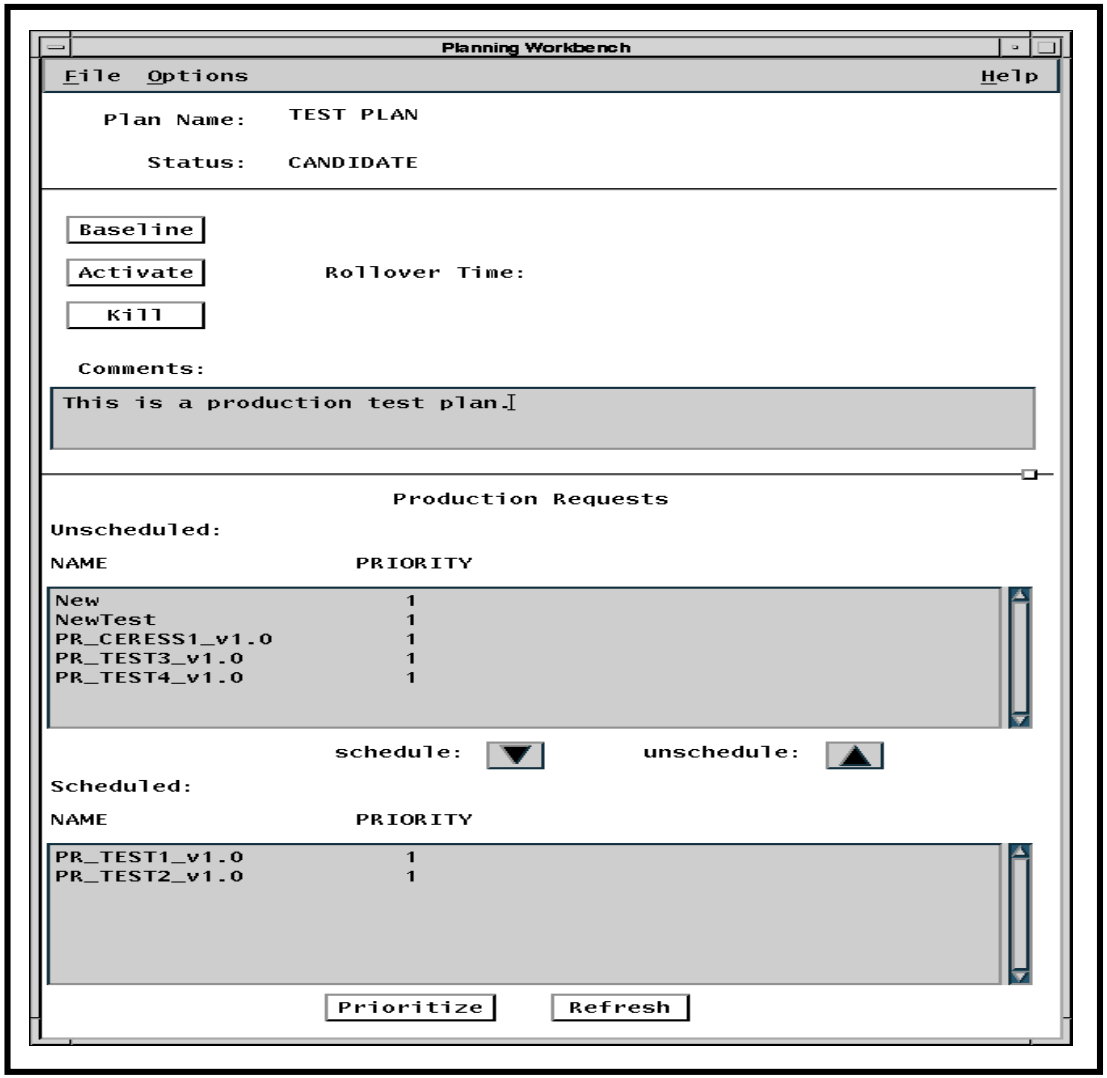
Creating a New Production Plan

telnet to (PDPS) p0pls01 or your PLN Host. Use login ID:, PASSWORD:, dce_login awhitele awhitele, setenv DISPLAY ….:0.0. setenv <mode>, and source EcCoEnvCsh, then follow instructions from Training Manual, Monitoring Production in PDPS Subsystem.

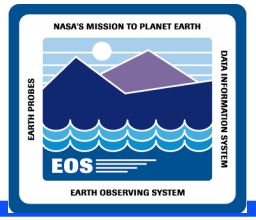
The Planning Workbench is launched to bring up the Planning Workbench GUI. The following information is entered:

- File - set to New. Plan Name - User name for plan.
- Status is CANDIDATE until plan is activated.
- Rollover Time - mm/dd/yy.
- Comment - Comments are optional.
- Production Request - Select one from the list of all PRs in the system.
- Schedule/Unscheduled - select PR and down arrow to schedule PR from unscheduled list and vice versa to unscheduled.
- Prioritize - Set priority.
- File - use “Save As”, to file under user specified PR name.
- Activate - Select to activate plan and enter date/time.
- Baseline - Select to create a new baseline plan.

Planning Workbench GUI



Review a Production Plan Timeline



Launch the Planning Workbench and view the production plans.

Select the File and Open.

- **This shows a list of Candidate, Active, and Baseline Production Plans.**

Select the desired Production Plan to view.

- **The timeline for the selected Production Plan is displayed.**

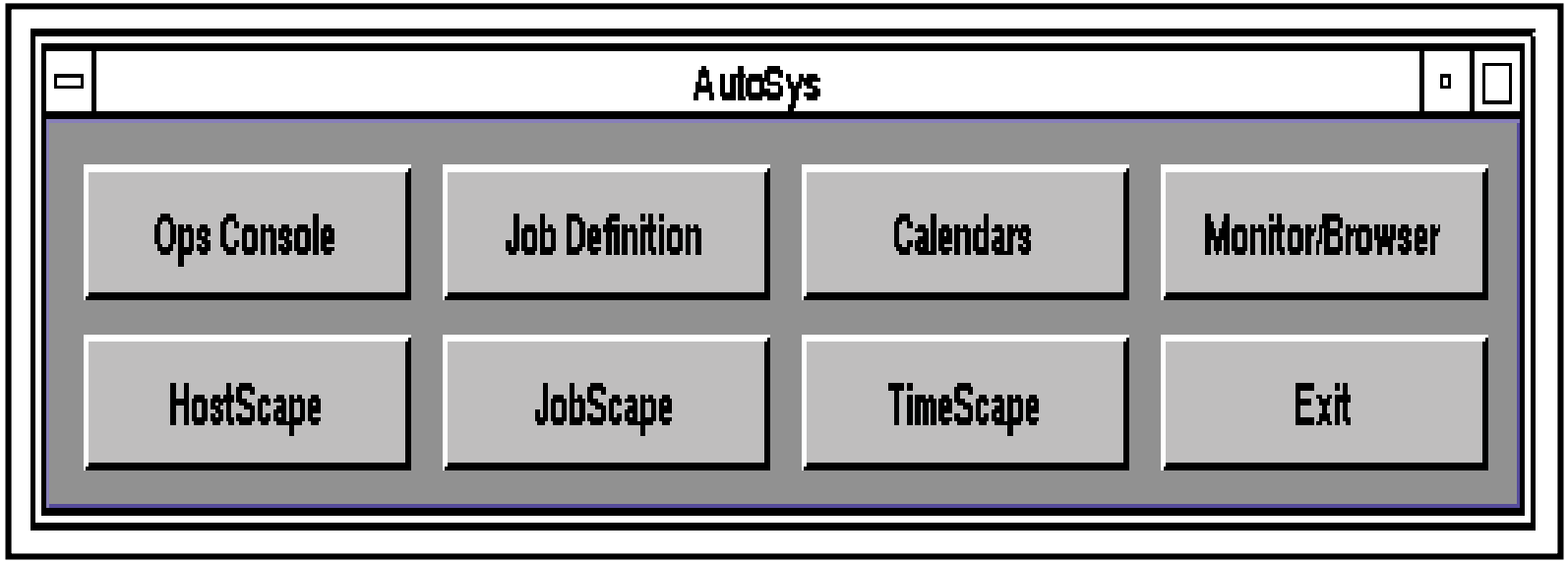
Monitoring Production

Monitor the PGE executions using AutoSys.

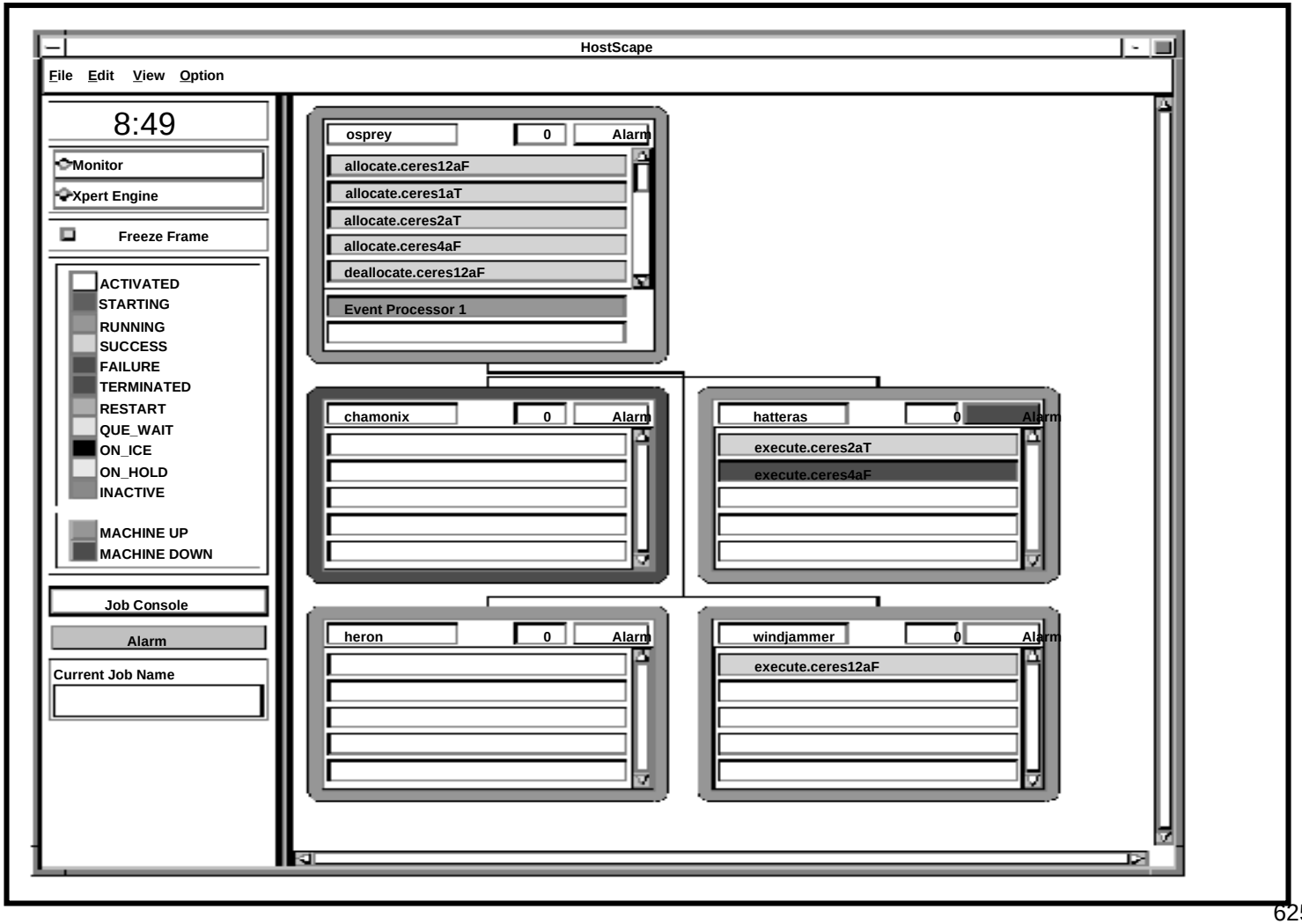
Process Steps:

- **Launch AutoSys Monitor from the SSIT Manager**
- **Select the DPRs to be displayed in the AutoSys Job Activity Ops Console Window.**
- **Select HostScape or TimeScape or JobScape for different views.**
- **View details of a single DPR.**
- **View the existing Event Report on the selected DPR.**
- **View processing alarms for a DPR.**
- **View job dependencies.**
- **Exit the AutoSys Monitor.**

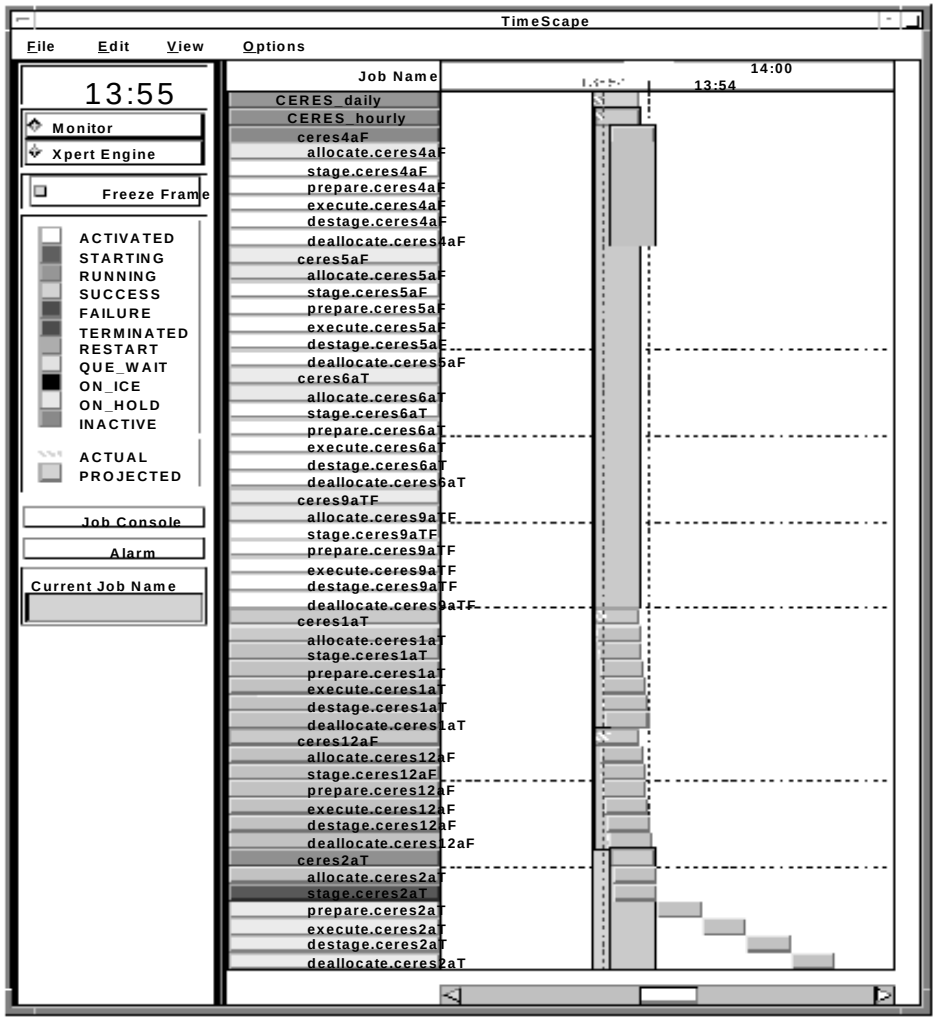
AutoSys GUI Control Panel



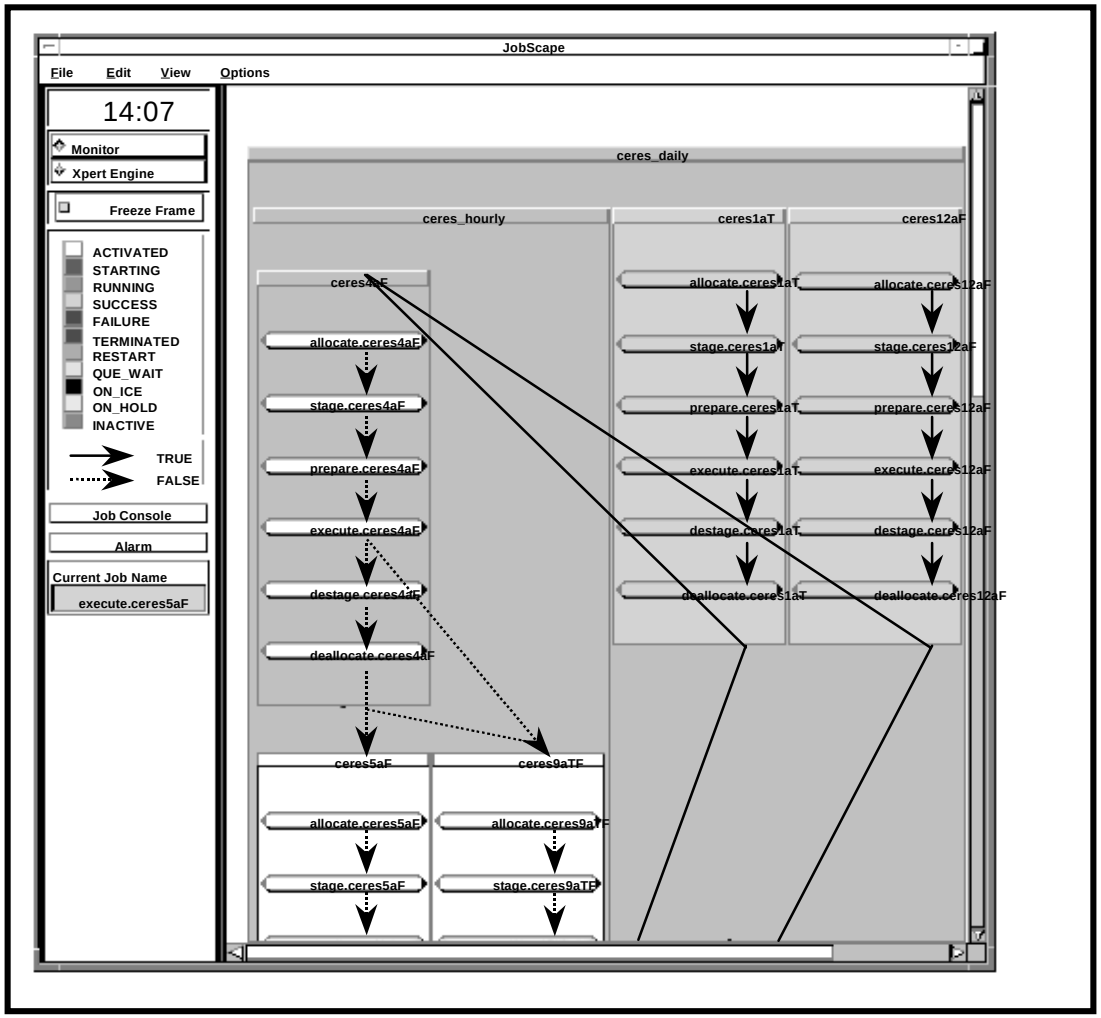
AutoXpert HostScape GUI



AutoXpert Timescape GUI



AutoXpert JobScape GUI

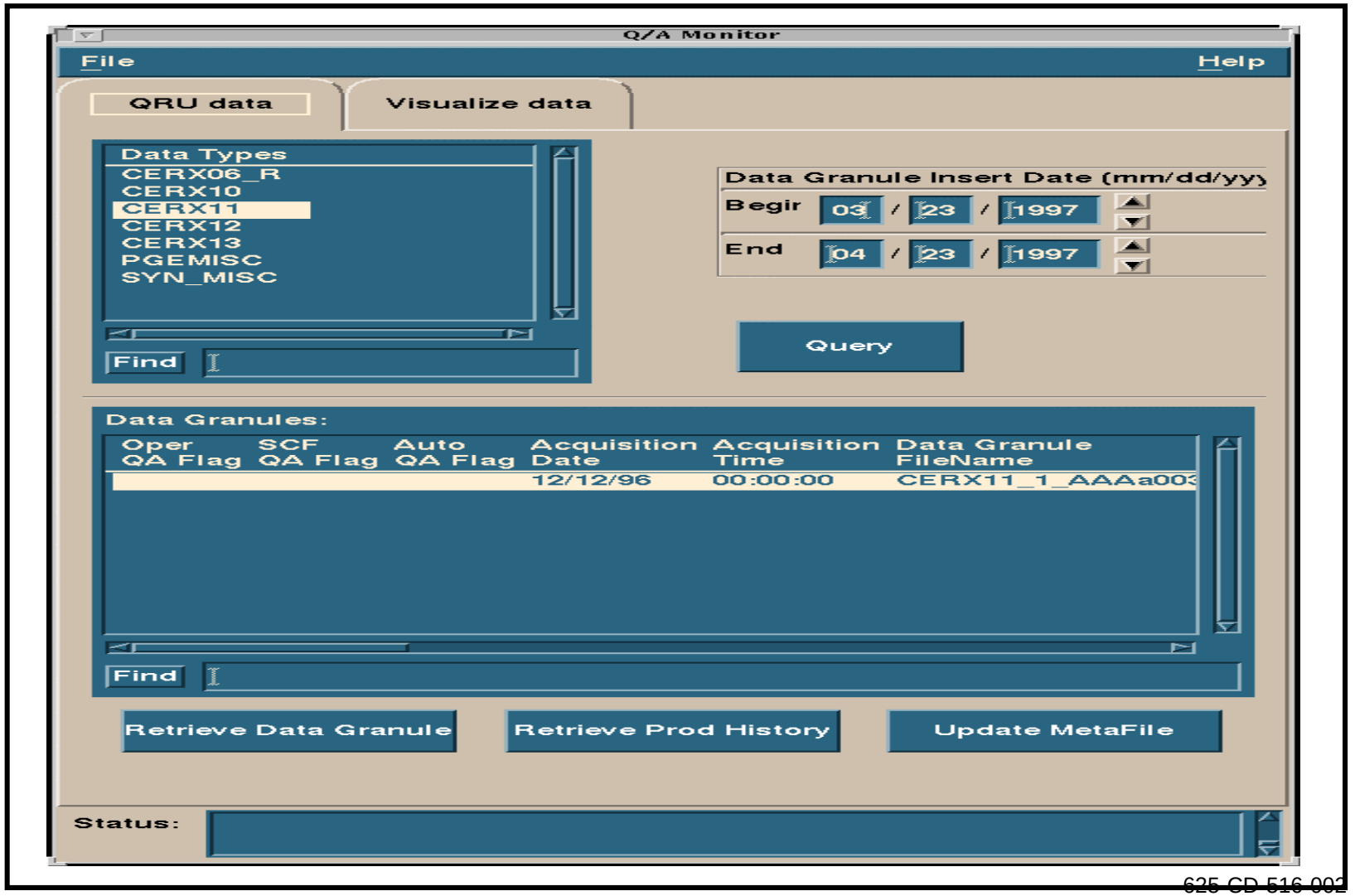


Post Processing Activities

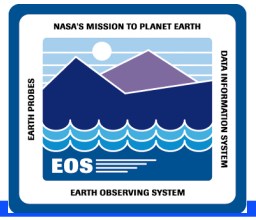
Invoke the QA Monitor to view the products and select options.

- Select Query data and select or query on ESDT and Duration (time range).
  - Select Data Granule(s) to check and select Retrieve Data Granule.
  - Production History Log files from PGEs run within PDPS.
  - Find Data Granule and select Retrieve Production History.
  - Examine Production History Log for metadata and other information.
- EOSView
  - Select Visualize Data.
  - Select Attributes to examine metadata in HDF headers.
  - Select Table to view data vectors and matrices.
  - If an image file is available, select Visualize.
  - From the SSIT Manager GUI → Tools →Product Examination→ File Comparison → EOSView or IDL.

Science Product QA



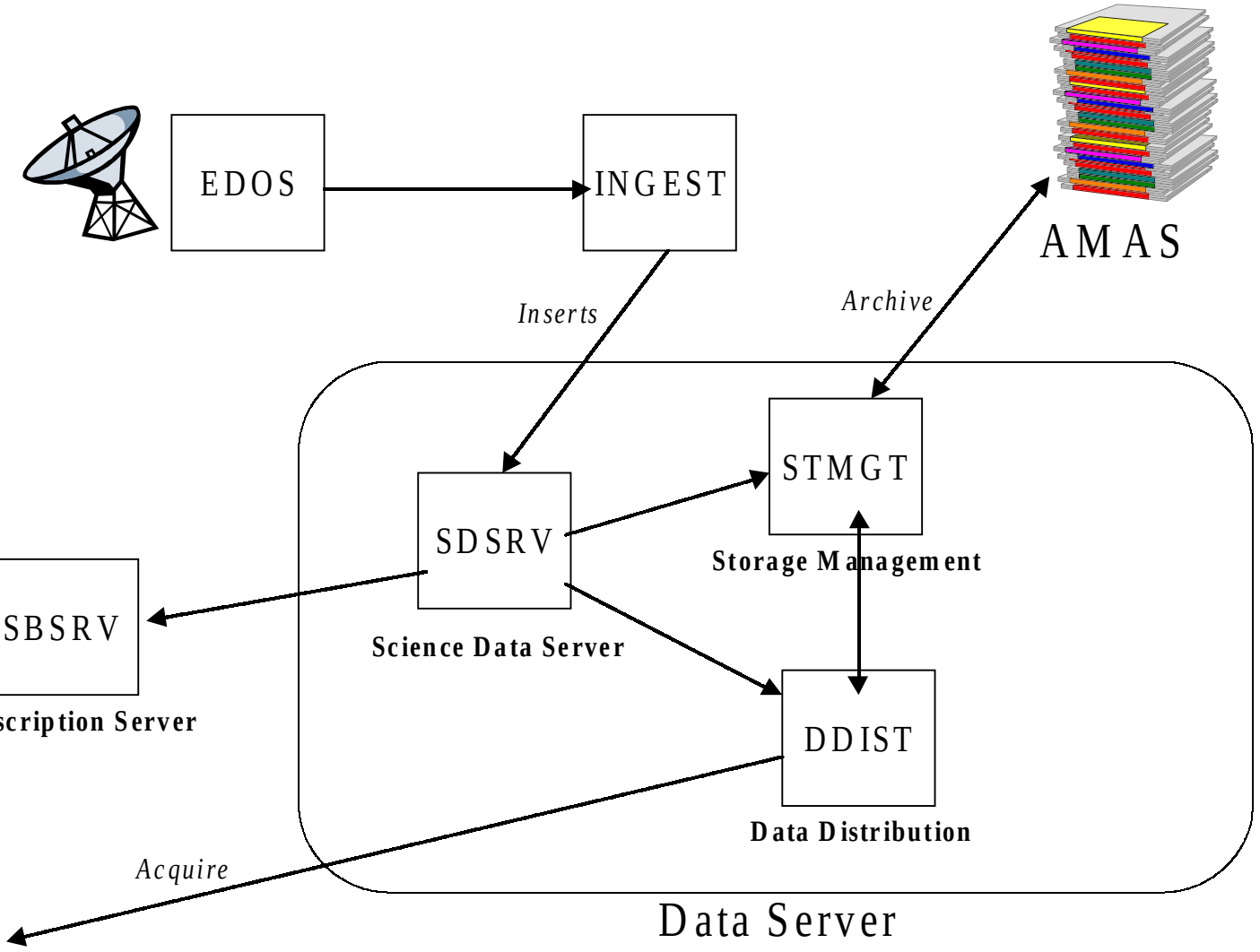
Postprocessing and General Investigation



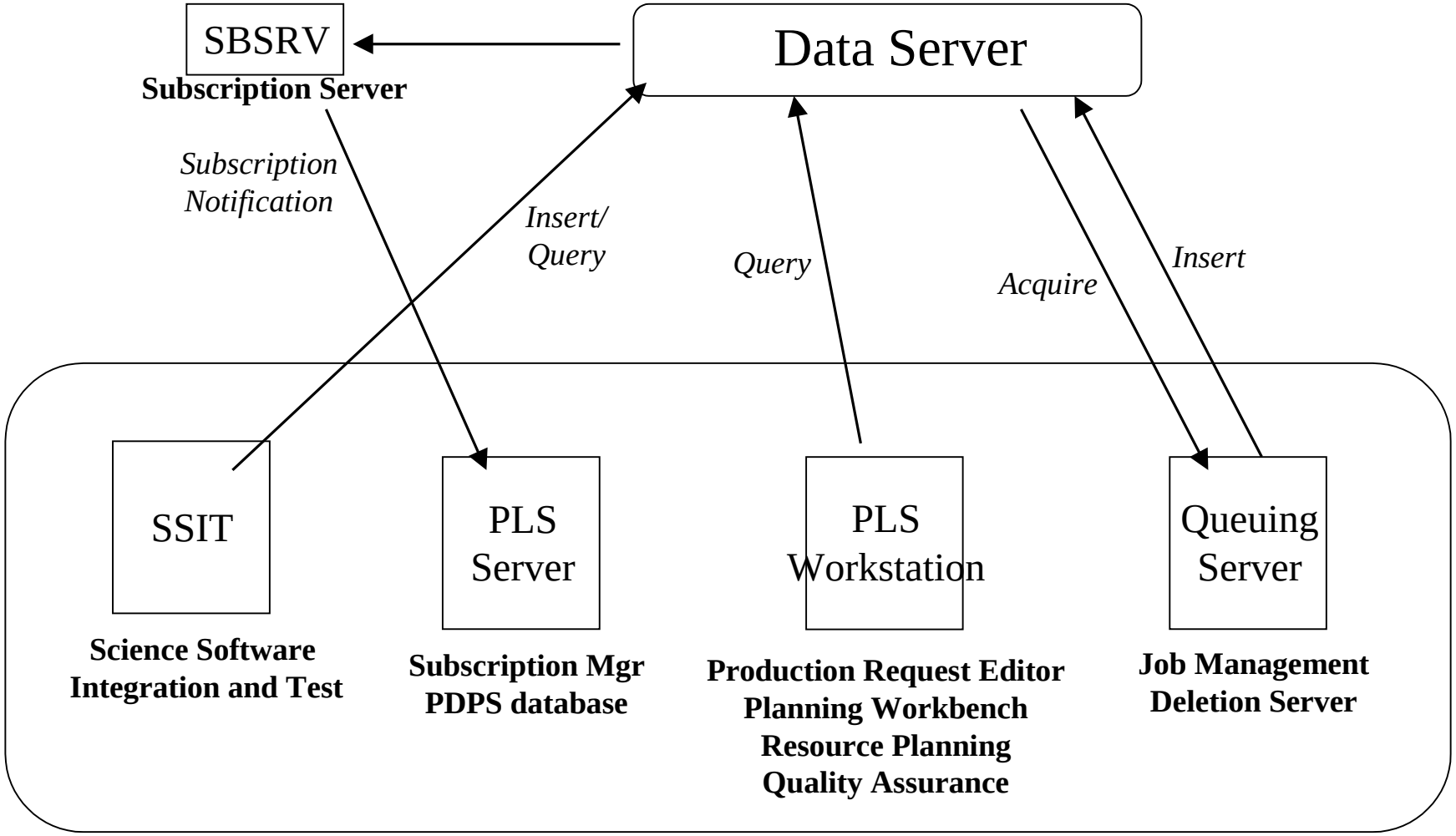
Detection of Science Software Problems and ECS Problems.

- **Examining PGE Log Files**
-
- **Investigate errors in Production History File.**
- **Use File comparison tools, EOSView and IDL to examine data product problems .**
- **Examine the PDPS Database with the Web Browser.**
- **Submit problems using DDTS.**

ECS/PDPS Overview



ECS/PDPS Overview cont.



PDPS - Planning and Data Processing System