611-TD-570-001

EOSDIS Core System Project

M&O Procedures: Section 26—SSI&T Operational Procedures

Interim Update

March 2000

Raytheon Systems Company
Upper Marlboro, Maryland

Preface

This document is an interim update to the Mission Operations Procedures Manual for the ECS Project, document number 611-CD-500-001. This document has not been submitted to NASA for approval, and should be considered unofficial.

This document has been revised to meet The Drop 5B Science System Release Plan.

Any questions should be addressed to Elroy McLeod or Rodney Creecy:

Data Management Office
The ECS Project Office
Raytheon Systems Company
1616 McCormick Drive
Upper Marlboro, Maryland 20774-5301

This page intentionally left blank.

26. SSIT Operational Procedures

The SSI&T operational procedures are given in this section. They are organized by activity. The order in which the procedures appear loosely follows the order in which they will usually be performed.

These procedures present the use of GUIs supplied in Release 4. Some procedures may have a command line equivalent; these are documented in the corresponding GUI help screens but are not presented here in the interest of simplicity. Version 2.0 Operations Tools Manual 609-CD-003-004 should be referred to for more detailed information on how to use GUI's and command line equivalent usage.

A Note about the Order of Procedures

The science software I&T operational procedures contained within this document are ordered. The order is intended to *loosely* suggest a logical sequence which, when used as a "road map", represents an overall, sensible end-to-end SSI&T activity at the Release 4 DAACs. The ordering cannot, however, be interpreted as a detailed, step-by-step guide to SSI&T activities. In addition, since there are many factors that affect the actual SSI&T activities during Release 4 (e.g. Instrument Team deliveries, DAAC policies, agreements between the Instrument Teams and the DAACs, etc.), the ordering in this document can only be suggestive.

Many of the procedures outlined in this document are inter-related. A procedure may assume that another procedure has been completed. In general, the ordering of the procedures reflects this. The user should be aware, however, that this is not the case for all procedures. Therefore, depending on the SSI&T activity, the ordering suggested may not apply. Procedures may require other procedures that appear *after* the procedure requiring them.

Assumptions

All procedures in this paper assume the following: that the Instrument Team has delivered the science software to the DAAC and that Release 4 ECS is available at the DAAC.

Conventions

The following conventions are followed for explaining procedures:

Text that should be typed literally in the "Action" column of the procedures is displayed in `courier` font. Text within a literal command that represents a fill-in-the-blank object is displayed in *`italic courier`* font. (Example: `cd` *`mydir`* means type "cd" and then type the name of the correct directory.)

A command line in the "Action" column that should be typed in without a line break will be indicated by an indent in any following lines. The end of the command is indicated by <ENTER>, which stands for pressing the ENTER or RETURN key.

26.1 Science Software Integration and Test (SSI&T) Preparation and Setup

26.1.1 Key Operator Roles

Science Coordinator: Provide support to Instrument Teams for the integration and testing of science software in the ECS system at the DAAC. Perform standard checking on all delivered software including source code, scripts, process control files and related documentation.

Science Data Specialist: Serves as a point-of-contact for planning, integrating, testing, and operating science software.

CM Administrator: Record, report, manage and distribute new and updated science software.

Science Software I & T Support Engineer: Provide support to Instrument Teams for the development, integration, test and problem resolution of science software.

Production Planner: Populate, maintain and schedule the production planning database for science software.

MODIS Science Data Processing Software Version 2.0 System Description Manual

This manual should be referred to for more detailed information on how to perform the SSI&T operational procedures as they apply to MODIS PGE's. It covers the specific attributes for each individual PGE and setup criteria.

26.1.2 COTS Software Tools

ClearCase: This tool is used as the ECS software configuration management tool. ClearCase provides a mountable file system which is used to store version-controlled data, such as source files, binary files, object libraries and spreadsheets.

Distributed Defect Tracking System (DDTS): This tool is used to electronically process configuration change requests (CCRs). DDTS will prompt the user for relevant information, identify the request and will mail these requests to pre-designated personnel.

26.1.3 General Process

The SSI&T process consist of two activities:

- **Pre-SSI&T Activity** - During this activity the Delivered Algorithm Package (DAP) is inspected, and tested in a non-production environment.
- **Formal SSI&T Activity** - During this activity, the Product Generation Executives (PGEs) are integrated with the DAAC version of the SDP Toolkit and executed on the ECS PDPS platform.

Key Terms:

- **Product Generation Executives (PGEs)** - The smallest scheduled unit of science software.
- **Delivered Algorithm Package (DAP)** - An ensemble of PGE source code, makefile, documentation, and other related files delivered in a package from the SCF to the DAAC for SSI&T..
- **Process Control File (PCF)** - Relate logical identifiers to physical files and other parameters required by the PGE.
- **Strings** - The processing hardware on which the science software runs.
- **Archive** - A File Storage Type indicating that granules that will be inserted Data Server are intended for long term storage and acquisition for distribution.
- **Collection** - A related group of data granules.
- **Granule** - The smallest data element which is identified in the inventory tables.
- **Product** - A set of output values generated by a single execution of a PGE for archival by ECS. A PGE may generate one or more products whose attributes are defined by the data provider.
- **Reliability** - Software reliability means that the software runs to normal completion repeatedly over the normal range of data inputs and running conditions.
- **Safety** - Software safety means that the software executes without interfering with other software or operations.

The science software in the DAPs will be integrated onto the PDPS and be used to produce the output data as determined by the algorithms. The refined and updated DAPs and data produced by the science software will eventually be provided to the subscribing user. Before the PGE is integrated into a production environment, extensive testing on the software must be performed.

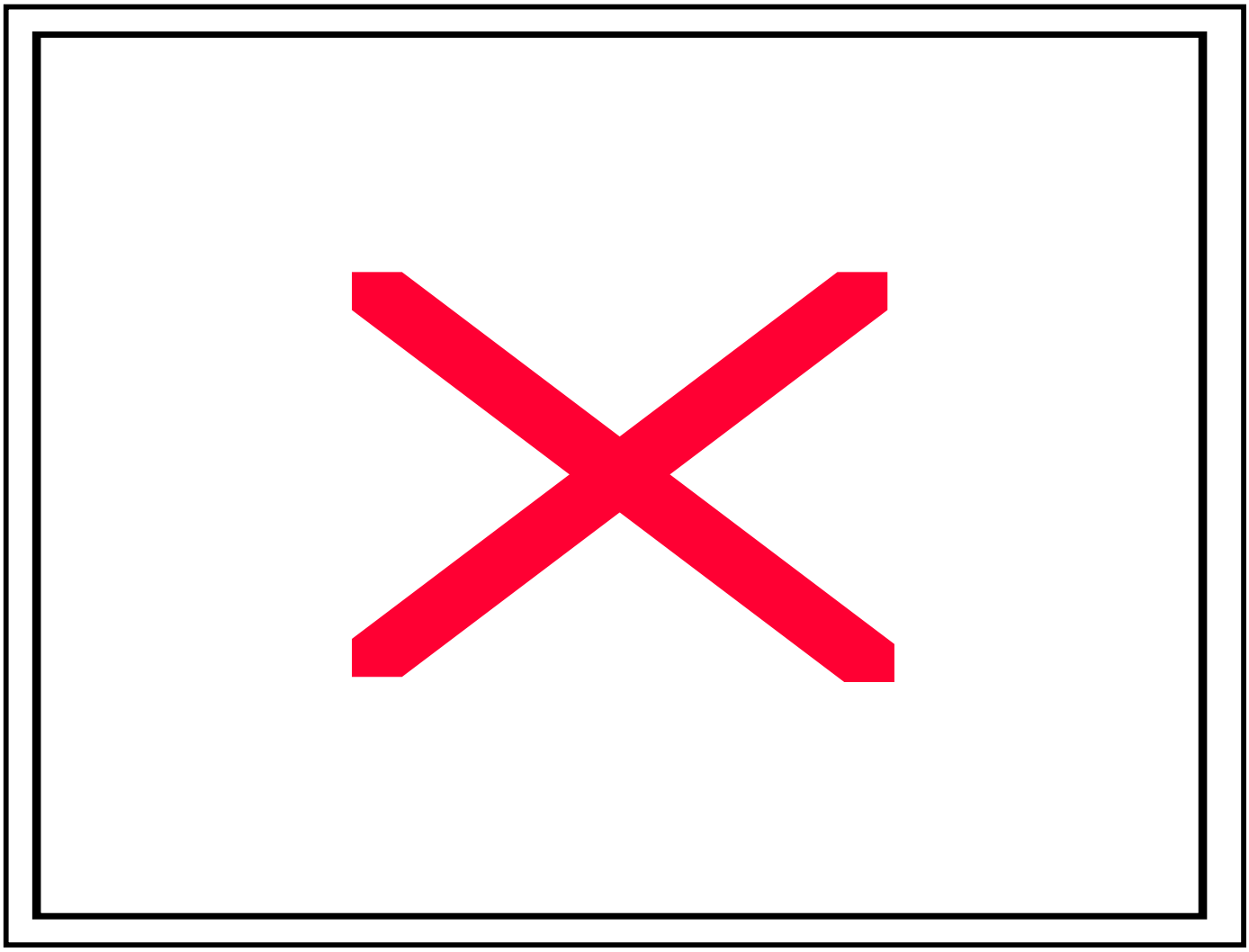
The following list provides a suggested, logical "road map" for getting science software tested and integrated into the ECS. This list is not intended to cover every situation and variations may be required.

26.1.3.1 GENERAL

- Science Software Integration and Test (SSI&T) is the process by which the science software is tested for production readiness in the DAACs in order to assure its (1) reliability and (2) safety. Prior to the delivery of the ECS software to the DAACs, SSI&T Checkout is conducted on early versions of the Products Generation Executives (PGEs) using separate system modes in the ECS Mini-DAAC, VATC (Verification and Acceptance Test Configuration), or the DAAC environments.

- SSI&T activities can be broadly separated into two categories: pre-SSI&T and formal SSI&T. Pre-SSI&T activities are those which do not involve the ECS Planning and Data Processing (PDPS) or the Science Data Server (SDSRV), but the formal SSI&T activities do involve the full ECS including the PDPS and the SDSRV.
- Most steps in the SSI&T process are inter-related and some steps may assume that another step has been completed. The ordering of the steps is very important but it cannot, however be interpreted as a detailed, step-by-step guide to SSI&T activities.
- Science Software Integration and Test consists of the following activities most of which are fully detailed in Science Software Integration & Test Operational Procedures for the ECS Project (162-TD-001).
- The activities described in the following list are also depicted in a (SSI&T Process Flow Diagrams 1 and 2) see **Figure** 26.1.3.1-1 and **Figure** 26.1.3.1-2. For a better quality depiction of these diagrams, refer to the Science Office Instruction No. SO-1-003 at the Web site listed in the next section below.

Figure 26.1.3.1-1 SSI&T Process Flow Diagrams 1



26.1.3.2 Science Office Project Instructions

The following are procedures listed from Science Office Project Instructions as they apply to the building of Science Data Products and SSI&T. A review of these documents is highly recommended at this time . They can be accessed from the Web using the following URL:

http://dmserver.gsfc.nasa.gov/proj_instr/sopi_index.html

Science Office ECS Project Instructions

Number	Subject	Issue Date
SO-1-002	Earth Science Data Type Generation Procedures PDF	5/11/98
SO-1-003	Science Software Integration and Test (SSI&T) PDF	9/14/98
SO-1-004	Science Office Science Support Internal Processes PDF,	
Science Office Science Support Internal Processes (161-IT-003-001)PDF		6/30/98
SO-1-005 Product Specific Attribute (PSA) Analysis PDF		7/23/98
SO-1-006	PGE Testing PDF	7/07/98
SO-1-007	Earth Science Data Types Testing and Integration PDF	7/24/98
SO-1-008	QA Metadata Update Tool (QAMUT) PDF	7/23/98
SO-1-009	Metadata Works PDF	7/23/98
SO-1-010 ECS Science Metadata Valids Update Procedures PDF		7/24/98
SO-1-011 Metadata-Process Established with MODIS PDF		8/20/98

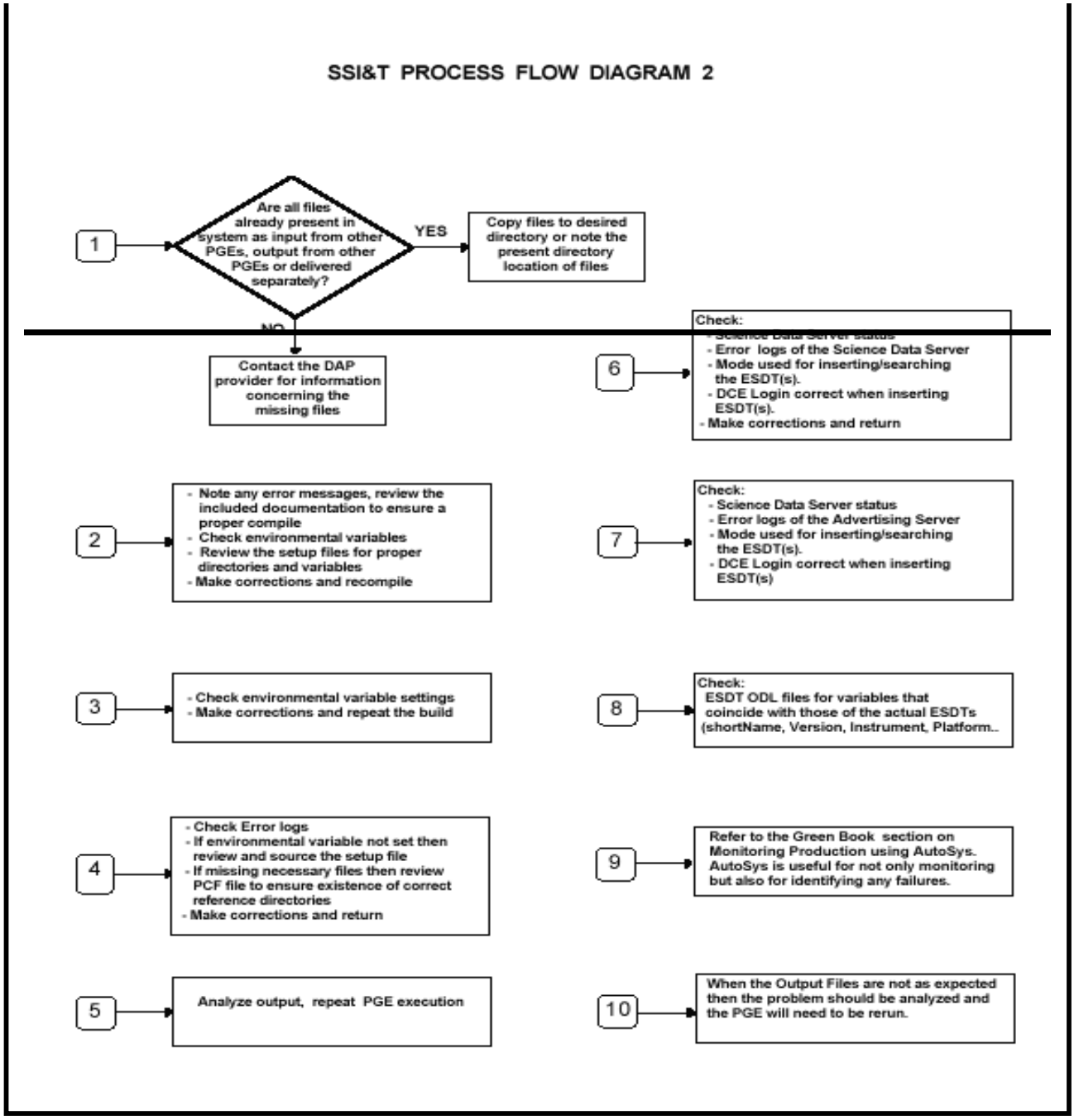


Figure 26.1.3.1-2 SSI&T Process Flow Diagrams 2

26.1.3.3 Pre-SSI&T Activities

1 As the DAP is delivered to a DAAC by the Instrument Team for SSI&T, the PGE listing documentation is reviewed.

2 The DAP is acquired and unpack and the documentation (i.e., packing list, readme, etc.) checked. The DAP contents are further checked by the Science Data Specialist to verify that

the contents match the packing list, agreed-upon directory structures are employed, location of files are correct, and all intended files and directories are present.

3 The Science Data Specialist requests that the CM Administrator place the DAP under Configuration Management control using ClearCase.

4 The SSI&T team checks the science software for standards compliance using the Process Control File Checker to check process control files (PCF), and the Prohibited Function Checker to check source files. Extract and check prologs.

5 The SSI&T team builds the science software into PGEs using the SCF version of the SDP Toolkit. Compile all source code. Link object code with appropriate libraries. . If the SMF files compile successfully, then proceed to Step 11 below; otherwise. the problem needs to be fixed and a successful compile must occur before proceeding further. This may require one or more of the following:

6 Note any error messages and review the included documentation to ensure a proper compile;

7 Check environmental variables;

8 Review the setup files for proper directories and variables;

9 Make corrections and recompile.

10 If the executable builds successfully, proceed to Step 12. If the build fails, it may necessary to do one or more of the following before proceeding:

11 Check environmental variables;

12 Make corrections and repeat the build.

13 Run the PGE from the Command Line.

14 If it the execution is successful, then the output files (products) are checked using the SSIT Manager file comparison tools; otherwise, one or more of the following needs to be done before proceeding:

15 Check error logs;

16 Check environmental variables;

17 Review and source the setup files;

18 If necessary files are missing, then review the PCF file to ensure the existence of correct reference directories.

19 The SSI&T team runs and profiles the PGEs from the UNIX command line on the SGI, saving the profiling results. They will be used later when entering operational metadata into the PDPS.

20 The SSI&T team collects performance statistics for the PGEs.

21 The SSI&T team examines the output log files from the PGE runs for any anomalous message. The SSI&T team compares the output product data with the delivered test data using the file comparison tools. If the products do not match the delivered test outputs

(expected outcome), the outputs should be analyzed and the PGE must be re-run. If the products match the delivered test outputs then

22 Steps 10 through 13 are repeated once using the DAAC Toolkit. If the products generated with the DAAC Toolkit match the delivered test output, formal SSI&T may begin.

23 SSI&T team reports any science software problems using the DDTS NCR process.

24 The SSI&T team reports any ECS problems using the DDTS NCR process.

25 The SSI&T team collects and logs all lessons learned.

26.1.3.4 Formal SSI&T Activities

1 For each ESDT used by the PGE, construct an ESDT ODL file for updating the PDPS or verify that they already exist. ESDT ODL files are also needed for all input and output data granules.

2 Construct a PGE ODL file for updating the PDPS database. This involves using the delivery PCF to construct an initial PGE ODL template file, which must then be hand edited to add required metadata. A mapping between logical IDs in the PCF and ESDT ShortNames must be known before this step is done.

3 Install ESDTs on the Science Data Server if verification indicates that they do not already exist. Installation links the PGE to all input and output ESDTs which allows the PGE to run within the PDPS. The Advertising Server must also receive notification of the update. If this fails then the ESDT's must be re-installed again after removing original ESDT's from the SDSRV. Note: While installing ESDT's the SDSRV intermittently coredumps. To clean-up you must remove the ESDT from ADSRV, SBSRV and DDICT and then try again.

4 The SSI&T Metadata is updated (PGE & ESDT Object Description Language or ODLs are created). This supplies metadata to the PDPS database

5 If the Metadata update is successful, then the Operational Metadata is updated; otherwise, the ESDT ODL files may have to be checked for correctness before updating the Operational Metadata.

6 Register the PGEs with associated data in the PDPS database. This step uses the PGE ODL from step 22 above.

7 For each input dynamic data granule needed by the PGE, construct a Target MCF and insert it to the Science Data Server.

8 For each input static granule needed by the PGE, construct a Target MCF and insert it to the Science Data Server.

9 Assemble the SSEP (as a tar file) and Insert it to the Science Data Server.

10 Initiate a Production Request (PR) that will result in one or more DPRs.

11 Use the Planning Workbench to plan the PR and hence, run the PGE.

12 Monitor the PGE run using AutoSys. The PGE's progress is monitored using the AutoSys COTS. The distinct steps that are visible on the AutoSys GUI and whose success is evident are Resource Allocation (.Al), Staging (.St), Pre-Processing (.Pr), Execution of the PGE (.EX), Post-processing (.Ps), De-staging (.Ds), and De-Allocation of resources (.Da).

13 If any of the steps in the execution is not successful, then each failure must be identified and corrected before proceeding to the next step.

14 Examine the output Production History File from the PGE runs for any anomalous messages. Compare the output product data with the delivered test data using the file comparison tools. . If any of the steps in the execution is not successful, then each failure must be identified and corrected before proceeding to the next step.

15 If the output files match the test output files and they are in Hierarchical Data Format (HDF), they are visualized using the EOSView tool, or the Interactive Display Language (IDL) tool. If the files are not HDF, then IDL is used.

16 Using the Planning subsystem, initiate more complex Production Requests if chaining is required.

17 Using electronic or hard media transfer methods, distribute the data products to the Instrument Teams for their review.

RECORDS

A weekly SSI&T status report is provided to NASA. This report contains the Performance Measurement Data.

PERFORMANCE MEASUREMENTS

SSI&T PGEs planned vs. actually delivered, pre-tested, and integrated is the metric used to monitor the effectiveness of the process described in the Procedure. Additionally, the Duration of Effort Required to Integrate in Work Days is used.

26.1.4 Preparation and Setup

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

Prior to the remote login, enter **setenv DISPLAY** <local workstation IP address>**:0.0** where the local workstation IP address represents the IP address you where you are located**.**

You may need to setup the terminal so that the remote host is displayed on your screen. (Sun machine) This is done by clicking on the **Application Manager** icon (the file drawer located at the bottom of the screen), followed by the **Desktop Tools** icon**,** followed by the **Terminal Console** icon

3 Perform a remote login by typing **rlogin [host]** then press the **Enter** key.

The **Enter Password** prompt is displayed.

4 Enter the **password** then press the **Enter** key.

5 Enter the directory where the setup script is located by typing **cd [**directory name**]** then press the **Enter** key.

6 Source the setup script by typing **source [**script name**]** then press the **Enter** key.

The setup script contains directory paths, sets of alias commands, and tools for SSI&T.

- For example, source the SSI&T script: Type **source /usr/ecs/{MODE}/CUSTOM/utilities /.buildrc <RETURN>**
  Note: This step only needs to be done once per login.
- source .buildrc may not be supported on a particular software drop. Therefore the SSI&T scripts will be built into other another script.

7 To ensure access to the multi server environment when needed, the following generic login commands have been established and should be used routinely:

- From a terminal: **xterm -n (host) &**
- From the xterm invoked: **telnet (host)**
- **login cmts1**
- **pw: ecsu$er**
- **dce_login DCE_user_name DCE_password and then press Enter Key.**
  DCE is an acronym for a Distributed Computing Environment.
- **setenv DISPLAY …..:0.0**

8 Listed are some of the GUI tools, typical servers (examples and their Host that need to be considered for activation when conducting SSI&T:

- **ECS Assistant, ADSRV/DM/IOS, p0ins02,**
- **ECS Assistant, SDSRV/DSS, p0acs03**
- **ECS Assistant, DPS, p0sps06**
- **ECS Assistant, SBSRV/CSS/IOS, p0ins01**
- **SSIT Manager tools, AITTL/DPS t1ais01**
- **Production Request, PLS, odyssey**
- **Planning Workbench, PLS, odyssey**
  Note: NETSCAPE should be closed to allow for a full screen GUI to be activated.
- **Monitor PGE, odyssey**

9 A second xterm should be activated with the same login procdures so as to monitor the (log files) when entering SSI&T files from GUI’**s.**

10 Servers can be brought down in any order. To bring them backup requires that they be brought up in **a sequential order to ensure connectivity**, the order is listed as follows:

- **STMGT, MSS, DDIST, IOS, SDSRV, PDPS**

11 The above servers have unique hosts assigned. Each host needs to be logged into the **generic login: ID, pw:, dce_login DCE_user_name DCE_password and then press Enter Key.** before activating ECS Assistant to carryout the downing and bringing up of servers assigned to their respective hosts.

26.1.5 SSIT Software Operating Instructions:

Starting the SSIT Manager GUI:

On workstation **x0ais##**, at the UNIX prompt in a terminal window, type as in step 1 below your user id and password.

NOTE: The **x** in the workstation name will be a letter designating your site:
g = GSFC, **m** = SMC, **l** = LaRC, **e** = EDC, **n** = NSIDC, **o** = ORNL, **a** = ASF, **j** = JPL; the ## will be an identifying two-digit number (e.g.,**g0ais01** indicates a Data Processing Subsystem (DPS) workstation at GSFC).

Prior to the rlogin, enter **setenv DISPLAY <local_workstation IP address>:0.0.** The **<ipaddress>** is the ip address of **x0ais##,** and **xterm** is required when entering this command on a Sun terminal.

---

1 **Example:** Log into an Algorithm and Test Tools (AITTL) environment using using a machine so configured. At the PVC this machine is **p0ais01.** A special host has been established using the **id**: and **password**:. Type: **setenv DISPLAY ….:0.0**

2 *Login to DCE* **(dce_login <DCE_user_name DCE_password and then press Enter Key.***>)*

What the user must do before trying SSIT functionality:

3 **setenv <mode>** : **(cd /usr/ecs/<MODE>/CUSTOM/utilities** Note that this only has to be done once per login.

4 This directory should contain scripts pertaining to setting the environment for SSIT Manager. Type in: **EcDpAtMgrStart <mode> &**

- This invokes the **SSIT Manager GUI** which should be displayed.

What must be done via SSIT tools:

Since SSIT is just a calibration of various tools, there is no specific order for which they must be run. Most tools can be brought up from the SSIT Manager GUI as well as started on their own.

The File menu provides the capability to exit the manager. The Tools menu provides access to the various tools that make up SSIT. The Run menu is customizable (allowing you to add your own scripts and tools) by editing the file *ssit_run_menu* in the *data/DPS* directory.

The checklist (first window on the GUI) allows you to check off various activities by double clicking on them. You may enter a commentary on the activity in the second window when

checking off a particular item. The file checklist.sample in the data/DPS directory can be edited to change the items in the checklist or its' location.

26.1.6 Updating the Leap Seconds and the Earth Motions files

The toolkit requires Leap Second and Earth Motion updates, weekly and twice weekly respectively, to accurately compute most time conversions. The following scripts have been established to accomplish these tasks as part of ECS support.

- **update_leapsec.sh**

This script updates the leapsec.dat file by ftp-ing to USNO and reformatting the information into the leap seconds file: $PGSHOME/database/common/TD/leapsec.dat
The present script, after obtaining the required file "tai-utc.dat" in the same Series 7 mentioned above, invokes PGS_TD_NewLeap, a C program that performs the actual update work. The function puts the current date in the header of the new leapsec.dat, with a remark that the file was either "Checked" (no new leap second) or "Updated" (new leap second). The date at which the USNO file used in the updating process was put on their server is also listed in the header.

- **update_utcpole.sh**

This script updates the **utcpole.dat** file on the basis of new data obtained by ftp to the U.S. Naval Observatory in Washington, D.C (USNO). Their data file is excerpted and the required fields are reformatted and written into the utcpole file: $PGSHOME/database/common/CSC/utcpole.dat

- **The Leap Seconds file:**

leapsec - file ID: **$PGSHOME/database/common/TD/leapsec.dat**
(Atomic time from International Earth Rotation Service)

Introduced every 12 to 24 months, announced almost 6 months in advance or as little as 90 days notice. Update available from U.S Navy Observatory (USNO).

Interval of update recommend: weekly, except Sundays 17:45 hours to 17:55 Eastern US time. Runtime is approximately 30 seconds**.**

- **The Earth Motion file:**

utcpole – file ID: **$PGSHOME/database/common/CSC/utcpole.dat**
(Record of the Earth's variable of slowing rotation with respect to UTC Time.)
Interval of update recommended: Twice weekly except Sundays 17:45 hours to 17:55 Eastern US time. Recommended scripts be run in the afternoon or evening each Tuesday and Thursday.

26.1.7 Script Name: update_leapsec.sh

The following processing tasks are carried out automatically by the use of this script:

- **Update via: Ftp to USNO, "maia.usno.navy.mil"** file accessed for leapsec**: tai-utc.dat. (Tests connectivity by using "ping')**

- **Function to be applied: PGS_TD_NewLeap,** excerpts and reformats the new information and appends new data and date to **leapsec.dat** file. A remark that the file was either "Checked" (no new Leap second) or "Updated" (new leap second). 26.2.2 Script Name: update_utcpole.sh

The following processing tasks are carried out automatically by the use of this script

- **Update via: Ftp to USNO, "maia.usno.navy.mil"** file accessed for utcpole**: finals.data. (Tests connectivity by using "ping')**

Function to be applied: PGS_CSC_UT1_update, excerpts and reformats the new information and appends new data to **utcpole.dat** file.

Guidelines:

1 The script must be run on a machine that has the Toolkit mounted and which can access the USNO site via ftp and access e-mail. (p0spg01 used at the Performance Verification Center (PVC))

2 For each installed Toolkit (including all modes, such as debug, F77, F90, etc.) the scripts need to be run only once, even if different platforms or operating systems are run. However, if entirely separate Toolkits exist at your installation, with different $PGSHOME home directories, then either the scripts need to be run in each, or the data files can be propagated from a primary Toolkit to the others.

3 It is highly desirable to have outgoing e-mail mounted on the machine of choice, so that error messages may be issued automatically from the scripts in case of failure.

4 If the updating process fails, then the script must be rerun. The Toolkit team should be contacted anytime the scripts are not giving the correct or accurate It is highly desirable to have outgoing e-mail mounted on the machine of choice, information. The 2 sets of scripts do also send an email message to SDP Toolkit mail address when a script fails

5 The Toolkit requires that the two data files not be too stale. Therefore the useful lifetime of the utcpole.dat and leapsec.dat files is 83 days. The Toolkit will issue an error message if no update was performed beyond 83 days. If this occurs you can expect geolocation accuracy to deteriorate to an extent that could require re-running for some of the more stringent users. If Tookit requires a leap second value after this date, an error message will be returned. This generally means that production will cease.

6 Keep the Latest files until your updates are completed! They are useful for a backup should they be needed.

Hardware Needed and Setup Procedures

The user's environment needs to be set up by running the script $PGSBIN/pgs-dev-env.csh or $PGSBIN/pgs-dev-env.ksh, depending on the shell being used. $PGSBIN stands for $PGSHOME/bin/mach, where "mach" stands for one of: sun5, sgi64, sgi, sgi32, ibm, dec, or hp. In other words it is a shorthand for the machine "flavor" you are using, and for sgi, the compiler option. Not all versions are necessarily at each DAAC or SCF, and in some cases the path may be more complicated. For example, at Goddard Space Flight Center DAAC, typical binary directories are /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/bin/sgi64_daac_f77/, or /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/bin/sgi64_daac_f90_debug/, for example.

Once the setup script is located and sourced, $PGSBIN is defined and your path includes it. Furthermore, a "PCF", or process control file, $PGS_PC_INFO_FILE is defined, which allows the executable functions invoked by the scripts to find the old data files, which are needed for the updates.

To run the scripts successfully, you must have write permission on the data files.

After the setup is done, just run the scripts. Both scripts (update_utcpole.sh and update_leapsec.sh) are located in the directory $PGSBIN, which will be in your path after the Setup script has been run.

On workstation **x0spg##**, at the UNIX prompt in a terminal window, type **source /data3/ecs/TS1/CUSTOM/daac_toolkit_f90/TOOLKIT/bin/sgi64/pgs-dev-env.csh .** This will set up the various environment parameters, such as PGSHOME, to enable the 64 bit version of the FORTRAN 90 compiler to be run.

NOTE: The **x** in the workstation name will be a letter designating your site:
g = GSFC, **m** = SMC, **l** = LaRC, **e** = EDC, **n** = NSIDC, **o** = ORNL, **a** = ASF, **j** = JPL; the ## will be an identifying two-digit number (e.g.,**g0spg03** indicates a Science Processor Subsystem workstation at GSFC).

Prior to the rlogin, enter **setenv DISPLAY <local_workstation IP address>:0.0.** The **<ipaddress>** is the ip address of **x0spg**

Example: To Update the Latest Leapsec.dat and Utcpole.dat files perform the following steps:

---

1 telnet to a machine that supports the Toolkit. (**telnet p0spg01**)
2 login: **ID**, **Password:**
3 **setenv DISPLAY ….:0.0**
4 **setenv PGSHOME /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit**
5 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/bin/sgi_daac_f77** then
6 **source pgs_dev-env.csh**
7 For **leapsec: cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/database/common/TD**
8 **cp leapsec.dat leapsec.dat_old**
9 Know thread for Leap Second run:

10 cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/src/TD then do **ls** – select: update_leapsec.sh or run script for Leap Second type in: **update_leapsec.sh**

A successful update will look like the following

P0spg01{cmops}[288]->update_leapsec.sh
Status of PGS_TD_NewLeap call was (0)
Status of MOVE command was (0)

1 For **utcpole:**
2 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/database/common/CSC**
3 **utcpole.dat utcpole.dat_old**
4 Know thread for utcpole run:
5 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/src/CSC** then do **ls** – select: update_utcpole.sh or run script for utcpole type in: **update_utcpole.sh**
6 A successful update will look like the following:

p0spg01{cmops}[294]->update_utcpole.sh
Status of PGS_CSC_UT1_update call was (0)
Status of MOVE command was (0)

26.2 Science Software Integration and Test (SSIT) Manager

26.2.1 SSIT Manager Overview

The principal tool used during SSI&T is the SSIT Manager. The SSIT Manager is the top-level graphical user interface (GUI) environment presented to SSI&T personnel. Its purpose is to bring together the tools needed for SSI&T into a single, graphical environment.

Across the top of the SSIT Manager are the toolbar items **File**, **Tools**, and **Run**. Clicking on each of these invokes a pull-down menu.

Under the **File** pull-down menu, the only item is **Exit**. Clicking on this causes the SSIT Manager to terminate.

The **Tools** pull-down menu has most of the SSIT Manager's tools. The menu items are:

- **Code Analysis** contains
- **SPARCwork** - A COTS package provided by Sun that allows for various coding activities including memory checking and debugging.
- **Office Automation** contains
- **MSWindows** - a Microsoft Windows emulator with MS Office (Word, Excel, PowerPoint) installed.

- **Ghostview** - for viewing PostScript formatted documents.
- **Netscape** - WWW browser and useful for viewing HTML formatted documents.
- **Acrobat** - for viewing PDF formatted documented.
- **DDTS** - for entering and tracking science software problems.
- **Standards Checkers** contains
- **FORCHECK** - for standards checking for FORTRAN 77 and Fortran 90 science software source code.
- **Prohibited Function Checker** - for checking science software source code for prohibited functions.
- **Process Control File Checker** - for checking Process Control Files (PCFs) delivered with science software.
- **Prolog Extractor** - for extracting prologs from science software source code.
- **Product Examination** contains
- **IDL** - Interactive Data Language tool supported by Sun. SSIT puts the user in the

IDL environment.

- **EOSView** - for viewing HDF and HDF-EOS files.
- **File Comparison** contains
- **ASCII** - for comparing two output products that are in ASCII format.
- **Binary** - for comparing two output products that are in binary format.
- **HDF (GUI)** - for comparing two output products that are in HDF or HDF-EOS format, GUI version.
- **HDF (hdiff)** - for comparing two output products that are in HDF or HDF-EOS format, command line tool.
- **Text Editors** contains
- **Emacs**
- **Xedit**
- **vi**
- **PDPS Database** contains
- **PCF ODL Template** - for converting delivered PCFs into ODL during PGE registration.
- **Check ODL** - for verifying ODL syntax of ODL files.
- **SSIT Science Metadata Update** - for updating the PDPS database with PGE information during PGE registration.
- **SSIT Opnl Metadata Update** - GUI for updating the PDPS database with PGE information during PGE registration.

- **Copy SSIT -> Production** - for copying PGE registration database information from SSI&T mode to Production mode.
- **Data Server** contains

- **Acquire DAP** - for acquiring a Delivered Algorithm Package (DAP).
- **Get MCF** - Source MCF is a Metadata Configuration File used to create a Target MCF (.met) for a Dynamic/Static Granule
- **Insert Static** - for inserting a static data file to the Data Server.
- **Insert Test Dynamic** - for inserting a dynamic test data file to the Data Server.
- **Insert EXE TAR** - for inserting a Science Software Executable Package (SSEP) to the Data Server.
- **SSAP Editor** - for editing and creating a Science Software Archive Package (SSAP) and inserting it to the Data Server.

The **Run** pull-down menu initially contains no menu items. Its purpose, however, is to allow a place for SSI&T personnel to place their own custom tools and scripts.

26.2.2 SSIT Manager GUI

This GUI (Figure 26.2-1) is the starting point for SSI&T activities. It provides access to a collection of tools that will be useful for this purpose.

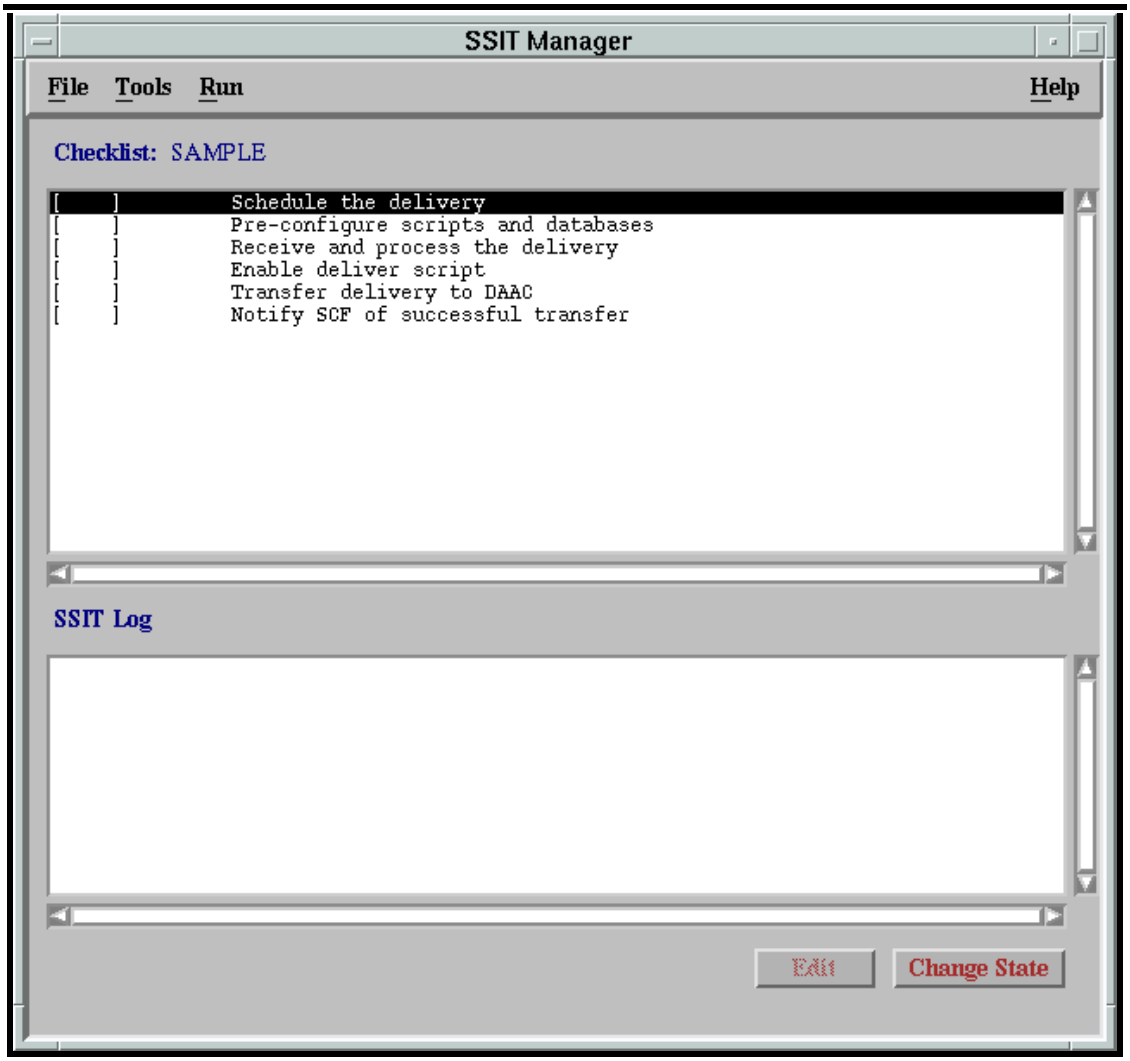


Figure 26.2-1. SSIT Manager Window

26.2.2.1 General Set Up of the SSIT Manager

The SSIT Manager requires a configured environment within which to run; it runs only on the AIT Suns. The set up steps described in this section need only be done the first time a SSI&T operator uses the SSIT Manager

To set up the environment for the SSIT Manager, execute the procedure steps that follow.

(This procedure was tested by **telnet p0ais01**, ID: **cmts1,** PW: **ecsu$er, setenv DISPLAY 155.157.123.34:0.0** or **setenv DISPLAY p0ais01:0.0** .

1 login to: dce_login DCE_user_name DCE_password and then press Enter Key.

2 setenv ECS_HOME /usr/ecs & setenv <mode>

3 **cp /usr/ecs/*mode*/CUSTOM/data/DPS/DpAtMgrInternal.pcf $HOME/*mySSITpcf***, press **Return**.

1. The ***mode*** is the ECS mode in which you are operating. This mode should be **TS1.**
2. The ***mySSITpcf*** is the file name of the private copy of the PCF that the SSI&T operator will use when running the SSIT Manager. The **$HOME** is the environment variable for the user's home directory. For example, **cp /usr/ecs/TS1/CUSTOM/data/DPS/DpAtMgrInternal.pcf $HOME/myPCF**, press **Return**.

4 At the UNIX prompt on the AIT Sun, type **setenv PGS_PC_INFO_FILE $HOME/*mySSITpcf***, press **Return**. (Check **env** for proper home path)

3. The ***mySSITpcf*** is the full path name to the private copy of the PCF to be used with the SSIT Manager when you run it (from step 1).

4. It may be useful to add this line to your .cshrc (or other start up script) so that it is set every time you login.

5 At the UNIX prompt on the AIT Sun, type **cd /usr/ecs/*mode*/CUSTOM/utilities**, press **Return**.

5. The ***mode*** is the ECS mode in which you are operating. This mode should be **TS1 or another mode assigned beforehand to operate in.**

6 At the UNIX prompt on the AIT Sun, type **EcDpAtMgrStart <mode> &**

- This invokes the **SSIT Manager GUI** which should be displayed.
- The checklist displayed within the GUI will be the default.

6. This sets environment variables and other settings needed for running the SSIT Manager.

26.2.2.2 Set Up of a Checklist for the SSIT Manager

The SSIT Manager offers the capability of maintaining user-defined checklist of SSI&T activities. The checklist is presented in the main window of the SSIT Manager. A default checklist is displayed unless a new checklist is specifically created. This procedure explains how to set up a customized checklist.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Creating a User-Defined Checklist for the SSIT Manager:

1 a From the SSIT Manager, click on the **Tools** menu, then choose **Product Examination**, then **EOSView**.

- The EOSView GUI will be displayed.

1 b Alternately, if EOSView isn't available from the SSIT Manager GUI, invoke EOSView from the command-line.

1 Go to the proper area by typing **cd /usr/ecs/TS1/CUSTOM/eosview** <RETURN>

2 Start EOSView by typing **EOSView** <RETURN>

2 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Open.**

- The **Filter** GUI will be displayed.

3 In the subwindow labeled **Filter**, enter full path name and file name wildcard template. For example, enter ***/home/MyDirectory/MySubdirectory/**.

- The ***/home/MyDirectory/MySubdirectory/**** represents the location to the directory containing the HDF-EOS files to examine.
- The asterisk (*) is a wildcard template that represents all files in that directory; other wildcard templates can narrow the search further, *e.g.* ***.hdf**.
- Use the **Directories** field to further select the correct directory.
- Files found matching the wildcard template in the chosen directory will be displayed in **Files** subwindow.

4 In the **Files** subwindow, click on the file name of the HDF-EOS file to examine. Then click on the **OK** button.

- A GUI labeled **EOSView - *MyOutputFile.hdf*** will be displayed where *MyOutputFile.hdf* is the file name of the file chosen in step 3.
- Be patient - this GUI may take some time to appear, particularly for large files.
- Once displayed, a list of HDF objects will appear in the main window. If nothing is listed, it means that no HDF objects were found within the file.

5 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on an object listed for which metadata is to be inspected.

- The object selected will be highlighted.
- Do not double click on object since this will cause a **Dimension** GUI to be displayed instead.

6 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **Attributes** menu and select **Global**.

- A GUI labeled **EOSView - Text Display** will be displayed.
- The global metadata associated with the object selected (in step 5) will be displayed in a scrollable field.
- If instead, the message “Contains no Global Attributes” appears, then the selected object contains no global metadata.

7 Repeat steps 5 and 6 for each HDF object within the selected HDF-EOS file for which metadata is to be examined.

8 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **File** menu and select **Close**.

- The **EOSView - *MyOutputFile.hdf*** GUI will disappear.
- Be patient - this GUI may take some time to disappear, particularly for large files.

9 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Exit**.

- The **EOSView - EOSView Main Window** GUI will disappear.

26.2.3 SSIT Manager Tools

There are several tools that are accessible through the SSIT Manager GUI. After selecting the TOOLS menu option of the menu bar, a set of options is available. See Figure 26.2.3-1, which indicates the use of the Tool menu item.

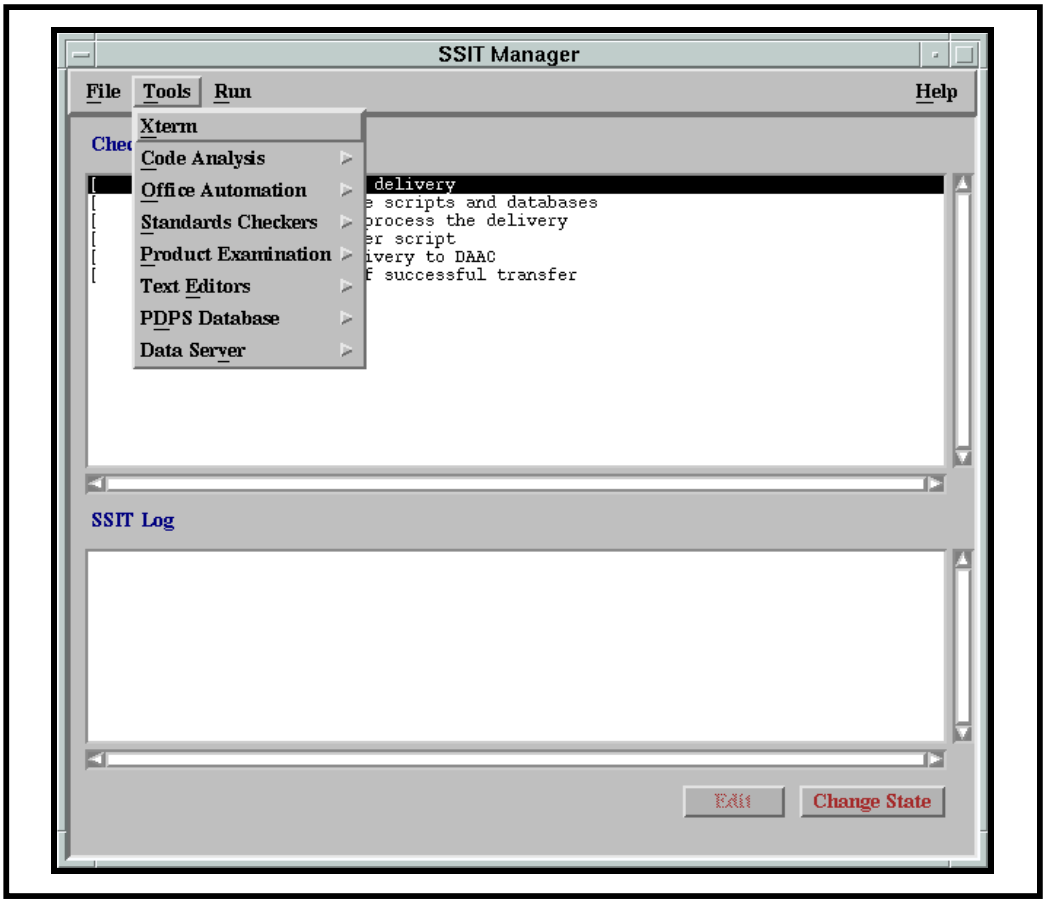


Figure 26.2.3-1. SSIT Manager Window - Tools Menu

26.2.4 Using the SSIT Manager:

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. The source file(s) are available, accessible, and have read permissions.
3. The below listed formatted text (ASCII) files containing the list of prohibited functions exist in the directory stored in the environment variable DPATMGR_DAT:

4. prohibitedFunctionsAda.txt
5. prohibitedFunctions.C++.txt
6. prohibitedFunctions.C.txt
7. prohibitedFunctions.F77.txt
8. prohibitedFunctions.F90.txt
9. If the source code files to be checked are in a VOB in ClearCase, a view has been set before the SSIT Manager was started.

26.3 Delivered Algorithm Package (DAP) - Acquiring, Unpacking, Subscription

The Delivered Algorithm Package (DAP) is the vehicle by which the PGE, source code, supporting files, documentation, etc. are delivered to a DAAC for SSI&T. Typically, the DAP is a compressed TAR file with a file name of form *string*.tar.Z . After initial processing, the DAP is broken apart into its components and those components will be subsequently processed and used based on their intended function.

The delivery mechanism for DAPs can be electronic (e.g. via UNIX ftp) or physical media (4 mm or 8 mm digital audio tapes).

26.3.1 Acquiring the Delivered Algorithm Package (DAP)

The following procedures are used by the SSIT team to acquire DAPs.

26.3.1.1 Acquiring the DAP via FTP

FTP is another method that the SSIT team uses in order to receive the science software. The following example demonstrates the FTP of the tar file from a remote machine.

Acquiring the DAP via FTP

---

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt, type **cd *DeliveryPathname***, then press the **Enter** key.

7. The ***DeliveryPathname*** is the full path name to the directory that has been set aside for ftp pull of DAPs from the Instrument Team. For example, **cd /home/*user*** where ***user*** is the user's login directory, then press the **Enter** key.
8. If the DAP is to be copied into a subdirectory, change to this subdirectory.

4 At a UNIX prompt, type **ftp *machineIPaddress***, then press the **Enter** key.

The ***machineIPaddress*** is the IP address or fully qualified domain name of the remote SCF machine. For example, **ftp 192.266.53.2**, then press the **Enter** key.

Or for example, **ftp aitg2sun.gsfc.ecs.nasa.gov,** then press the **Enter** key. The remote machine will likely display some messages and then prompt for a login name.

An ftp session is established.

5 At the ftp prompt on the remote machine, enter user login name, then press the **Enter** key.

The remote machine will typically respond with **331 Password required for *username*:**

6 At the ftp prompt on the remote machine, enter user password, then press the **Enter** key.

9. The remote machine will typically respond with **230 User *username* logged in** and display the **ftp>** prompt for further ftp commands.

7 At the ftp prompt on the remote machine, type **cd *DAPpathname*** then press the **Enter** key.

10. The ***DAPpathname*** is the full path name to the directory on the remote machine containing the DAP to retrieve. For example, **cd /home/mac ,** then press the **Enter** key. The directory location should be known.

8 At the ftp prompt on the remote machine, type **binary**, then press the **Enter** key.

11. The **binary** command causes subsequent file transfers to be in binary mode, preserving the integrity of the file to retrieve without interpretation (as would be done in ASCII mode.
12. The system will typically respond with the message **200 Type set to I** indicating that binary mode has been set.

9 At the ftp prompt on the remote machine, type **get *DAPfilename***, then press the **Enter** key.

13. The ***DAPfilename*** is the file name of the DAP to retrieve.
14. • For example, type **get TestPGE.tar**, then press the **Enter** key.
15. The user may need to type **dir** then press **Enter** to display a listing of the files in the current directory. The system will likely display several lines of messages once the transfer has completed. For large files, this may take a long time (minutes to hours depending upon the size of the DAP and the bandwidth of the connection).

10 At the ftp prompt on the remote machine, repeat step 9 or type **quit**, then press the **Enter** key.

16. Typing **quit** and pressing **Enter** closes the ftp connection with the remote machine.
17. Retrieve other DAP files by repeating step 9. The DAPs retrieved will reside in ***DeliveryPathname*** on the local machine.

11 At the UNIX prompt type **cp /home/mac/TestPGE.tar**, then press the **Enter** key.

18. This step will copy the DAP tar file into their working directory.

26.3.1.2 Acquiring the DAP from the Archive after Ingest

The **insert** service is used to put the DAP into the Data Server after it is ingested. Once the DAP is in the Data Server, the **acquire** service is used to retrieve it.

DAP is acquired from Data Server and placed in the specified directory. Note there will be 2 files, the DAP itself (a big tar file) and the metadata associated with the DAP. The metadata may be helpful in the creating the SSAP.

When a DAP is inserted into the Data Server by Ingest, an email is sent to all users who subscribe to that event (Section 26.3.2).

26.3.1.3 Performing a DAP Acquire Using SSIT Manager

Generally, the preferred approach to accomplishing a DAP **acquire** will be through the use of the SSIT Manager GUI.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The following servers/services are up and operational:
   Data Server, Subscription Server, Storage management
2. The following must have occurred between those Servers/Services:
3. Ingest must have ingested DAP and Inserted it into the Data Server. Subscription Server must have gotten notification from the Data Server of the Insert. Subscription Server must send email to the SSIT operator notifying him/her of DAP Insertion and giving him (in the email) the UR of the DAP.
4. The SSIT Manager is available
5. The X Window **DISPLAY** environment variable is pointing to your screen

DAP Acquire Procedures:

1 If not already on an AIT Sun, log onto one from your current machine.

2 Bring up the SSIT Manager GUI. At the UNIX prompt, type **mgr** (if alias has been established)

3 After a short while, the SSIT Manager GUI will appear. From the SSIT Manager top menu bar, select **Tools -> Data Server -> Acquire DAP**
See figure 26.2.3-1. If the SSIT Manager GUI is used to initiate the DAP processing, Step 4 can be skipped.

4 Alternately, one can initiate the DPA processing sequence from the command line. To do this

5 Type **source /usr/ecs/TS1/CUSTOM/bin/DPS/.buildrc <RETURN>**
Note: This step only needs to be done once per login

6 Type **/usr/ecs/TS1/CUSTOM/bin/DPS/DpAtStageAlgorithmPackage.sh <RETURN>**

7 The user will be prompted with:

```
        ** DAP Staging Tool **
Configuration filename? (enter for default: DpAtAA.CFG)
```

To respond, type **<RETURN>**

8 The user will be prompted with:

ECS Mode of operations? (enter for default: OPS)

To respond, type **TS1 <RETURN>**

9 The user will be prompted with:

Name of email message file (including path)?

To respond, type the required file name plus the path, e.g.,
/home/diascone/emessage01.asc <RETURN>

The user will be prompted with:

Directory to receive staged file?

To respond, type the required directory, e.g.,
/home/diascone/staged <RETURN>

26.3.2 Unpacking a DAP

Once a DAP has been acquired via electronic means or physical media, it typically needs to be unpacked before its contents are accessible for SSI&T. Several mechanisms are available under standard UNIX for packing and unpacking files to and from a file archive, the most common being UNIX *tar*. Another fairly typical utility is *gzip* and its companion, *gunzip*.

The file name extension is usually an indication of the packing utility used and DAP files should use this convention. DAP files that have been packed using the UNIX *tar* utility will usually have .tar as a file name extension indicating a tar file. If the DAP has been further compressed using the UNIX *compress* utility, the file name extension is typically .tar.Z indicating a compressed tar file. For DAP files packed with the *gzip* utility, the .zip file name extension is generally used.

When unpacking is performed on a DAP, the contents of the packed file are moved from the tar archive to local disk. If the DAP tar file contains directories as well as files, these directories will be created in the same structure as in the tar file. This structure typically reflects the directory structure from which the tar file was created in the first place at the SCF. Once a tar file has been unpacked, the original tar file will still exist unaltered.

Unpacking a DAP

3 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.
3 Enter the **password** then press the **Enter** key.
3 At a UNIX prompt, type **cd *UnpackPathname***, then press the **Enter** key.
3 The ***UnpackPathname*** is the path name of the directory that has been set aside for unpacking of DAPs.
3 This directory now contains the DAP tar file. For example, **cd /home/*user***, where ***user*** is the user's login directory, then press the **Enter** key.
3 If the tar file is compressed, at a UNIX prompt, type **uncompress *PackedDAP.Z***, then press the **Enter** key.
3 The ***PackedDAP.Z*** is the file name of the compressed DAP file.
- The file name extension of **.Z** is a convention indicating UNIX compressed files. The **uncompress** utility expects this file name extension by default. A resulting error may indicate that the DAP file was not compressed or that another compression utility was used. If the file name extension was **.Z**, the uncompressed version will have the same file name but without the **.Z**, for example ***PackedDAP***.
- The tar file for the SSI&T Training will not be compressed.

3 At the UNIX prompt, type **tar xvf *PackedDAP***, then press the **Enter** key.
- The ***PackedDAP*** is the file name of the uncompressed DAP file.
- The tar archive will be unpacked in the current directory. If the archive contained directories and subdirectories, these will be created by the tar utility and populated by the files that belong.

26.4 Science Software Configuration Management

The CM Administrator and System Administrator are key players in the SSI&T process. The CM Administrator receives the science software from the Science Data Specialist, places these files into a directory and request that the System Administrator place the files under configuration control by using the ClearCase tool. The science software is then tested by the SSI&T team and once the science software has successfully been tested, and upon direction from the CCB, the files are distributed to the Production Planner for placement on production server.

The CM and System Administrator need a good understanding of the ClearCase tool. ClearCase will be used to create a view, create a new directory, import files into the temporary subdirectories, and check-in and check-out files.

26.4.1 ClearCase Overview

All data managed under ClearCase are stored in Versioned Object Bases (VOBs), which are the "public" storage areas and Views, which are the "private storage areas. VOBs are data structures that can only be created by the CM administrator using the mkvob ("make vob") command. A

VOBs is mounted as a file system and when viewed through a view, it appears as a standard UNIX directory tree structure. This file system, accessed through its mount point, has a version-control dimension which contains file elements and versions of file elements. Once reviewed, the System Administrator will place these files under configuration control. In order to accomplish this task, a view must be created in ClearCase. A view is necessary in order to make visible and accessible files and directories that have been checked in to a VOB.

Data that are under configuration management in ClearCase are said to be "checked in". In order to alter a checked-in data element (e.g. a file) to make a newer version of it, the data element must first be "checked out". Once the change has been made to the checked- out version, it is checked in again. The VOB will then contain both versions of the data element and either can be retrieved at a later date.

In general, executable binary files, object files, and data files should not be checked into ClearCase. Binary and object files are not stored efficiently in ClearCase; data files for software may be extremely large and a VOB is typically not sized for this.

Files that should be checked into ClearCase include source code, scripts, makefiles, assorted build and run scripts, documentation and other ASCII files.

A Versioned Object Base is defined by the following characteristics:

A mountable file system which stores version-controlled data, such as source files, binary files, object libraries, WYSIWYG documents, spreadsheets and anything which can be stored in the UNIX file system.
Can be mounted on some or all workstations
Several VOBs may exist on a machine or on different machines on a network.
When mounted as a file system of type MFS, a VOB can be accessed with standard UNIX and ClearCase tools.
The ClearCase file system is transparent.
Created by the CM administrator

A VOB is comprised of:

Storage area for versioned files, derived objects and cleartext files.
Database (live, shadow and log file).

26.4.2 Creating a View in ClearCase

In order to make files and directories that are in a ClearCase VOB visible and accessible a ClearCase view must set. A ClearCase view need only be created once. Once created, the view can be set at the beginning of each user session. Multiple views for a single user may be created.

In order for the SSI&T tools under the SSIT Manager to have access to the ClearCase VOB, the ClearCase view must be set before the SSIT Manager is run.

A view is defined by the following characteristics:

A working context for an individual developer or closely coordinated group.
Can be used to access any VOB or multiple VOBs.
Selects versions of VOB directories and files to display.
Allows developer to work without interfering with other developers.
Not a set of files but a way of seeing shared elements.
Each user may have multiple views for new development, bug fixing or porting activities.

A view is comprised of:

View storage area (typically in a local machine) - private storage for checked-out files, derived objects and private files.
Configuration Specification - set of rules which determine the version of a file the view will see.
View-tag - Name given to the view (ex. angies_view), view-tags are registered in /urs/adm/atria/view_tags.
Objects stored in a view:
Checked-out versions of file elements.
Unshared derived objects.

The ClearCase procedures can either be run from the UNIX command line or from the File Browser Screen. The SSI&T Training will only cover the UNIX command line procedures. The corresponding GUI procedures are included in the Training Material for future reference.

The following procedure not only will create a view, but will also allow creation of a subdirectory where new science software files may be stored.

Assumptions:

1. ClearCase is available.
2. A Versioned Object Base (VOB) has been created

26.4.2.1 Creating a View in ClearCase Using Command Lines

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt type **cleartool lsview**, then press the **Enter** key.

19. The **lsview** command displays the pathname to the storage location of the views.

4 At a UNIX prompt type **cleartool mkview -tag *ViewName ViewPath/ViewName*.vws,** then press the **Enter** key.

The ***ViewPath*** is the full path to the directory where views are stored.

The ***ViewName*** is the user selected name for the view. The file name for the view must end in "**.vws".**

For future reference, the corresponding ClearCase GUI procedures are included in the following section.

26.4.2.2 Creating a View in ClearCase using the File Browser Screen

Selecting a view listed in the View Tag Browser screen brings up the File Browser, or main screen, shown in Figure 26.4.2-1.

Displays the directory name of the current VOB, just below the toolbar.
Displays the content of the directory in the space below the directory's name.

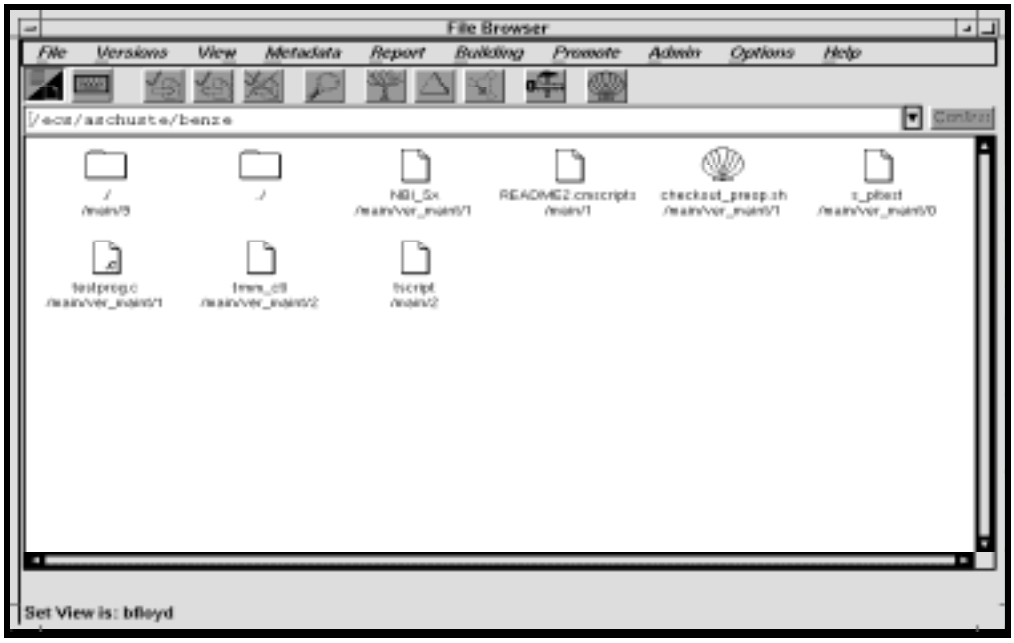
- 

Figure 26.4.2-1 ClearCase File Browser Screen (Main Screen)

Procedures

1 The user should log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.
Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 Invoke ClearCase by typing **xclearcase &** on the UNIX command line then press the **Enter** key.
- The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase **View Tag Browser** screen is displayed listing available views.

4 To create a view for checking in the software change package, select a known View and press the **Enter** key.
- The File Browser window is displayed.

5 Select **File→Execute→Single Command.**
- The String Browser window is displayed.

The prompt Enter shell command to run is displayed.

6 Invoke the make view command by typing **mkview [filename]** on the UNIX command line and press the **Enter** key.
- The **tempdisp** window appears.

- The **View [filename] Created Successfully** and the **Cache Updated for View [filename]** prompts are displayed.

7 Close the **tempdisp** window by clicking on the window and press the **Enter** key.

- The **tempdisp** window closes.

8 Select **View →List** from the menu.

- The **View Tag Browser** is displayed.

9 Find the new view by scrolling through the list until the new view is observed.

26.4.3 Setting a View in ClearCase

In order to make files and directories that are in a ClearCase VOB visible and accessible, a ClearCase view must be set. Only one view can be set (active) at a time.

26.4.3.1 Setting a View in ClearCase Using Command Lines

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 At a UNIX prompt type **cleartool setview *ViewName*** where ***ViewName*** is the user's view created in the previous section, then press the **Enter** key.

26.4.3.2 Setting a View Using the File Browser Screen in ClearCase

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 Invoke ClearCase by typing **xclearcase &** on the UNIX command line then press the **Enter** key.
The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
The ClearCase **View Tag Browser** screen is displayed listing available views.

4 To set a view, select a known View and press the **Enter** key.
The File Browser window is displayed.

5 Select **File→Execute→Single Command.**
The String Browser window is displayed.
The prompt **Enter** shell command to run is displayed.

6 Invoke the set view command by typing **setview ViewName** on the UNIX command line and press the **Enter** key.
ViewName is the name of the view to set.

26.4.4 Creating a New Directory

In cases where a new directory needs to be created and placed in ClearCase, the user will activate ClearCase and create a new directory. This type of procedure is necessary only if a new directory is required.

The following is a list of tools, and or assumptions:

1. A VOB has been created at the UNIX directory.
2. A view has been created.

- **Creating a New Directory in ClearCase Using Command Lines**

---

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt type **cleartool setview *ViewName***, then press the **Enter** key.

The ***ViewName*** is the user's view.

4 At a UNIX prompt type **cleartool lsvob,** then press the **Enter** key.

This command lists all the VOBs and allows the identification of the SSI&T VOB.

5 At a UNIX prompt type **cd *pathname***, then press the **Enter** key.

The ***pathname*** is the full path name of the parent directory in the VOB in which the new directory is to be added.

6 At a UNIX prompt type **cleartool checkout -nc .** then press the **Enter** key.

This command checks out the current directory. Note the dot for the directory.

The **-nc** is a keyword used when no comments are to be made for this action.

7 At a UNIX prompt type **cleartool mkdir -nc *dirname,*** then press the **Enter** key.

The ***dirname*** is the name of the new directory being created.

8 At a UNIX prompt type **cleartool checkin -nc *dirname,*** then press the **Enter** key.

This command checks in the new directory named ***dirname***.

9 At a UNIX prompt type **cleartool checkin -nc .** then press the **Enter** key.

This command checks in the current directory.

26.4.4.2 Entering a New Directory Using the File Screen Browser into ClearCaseÄ

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 Invoke ClearCase by typing **xclearcase &** on the UNIX command line then press the **Enter** key.

- The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase **View Tag Browser** screen is displayed listing available views.

4 Select **File→Execute→Single Command.**
The String Browser window is displayed.
The prompt **Enter shell command to run** is displayed.

5 Invoke the make directory element by typing **mkdir [filename]** on the UNIX command line and press the **Enter** key.

6 Invoke the make element command by typing **mkelem [directory name]** on the UNIX command line and press the **Enter** key.

7 Type into the directory input box of the **File Browser** the name of the directory in the VOB to be checked out, press the **Enter** key, then follow the menu path **Version→Checkout→Reserved: no comment**.

- In order to add new files to ClearCase, the directory in which the files are to be added must be checked out first.
- ClearCase forces the checkout onto a maintenance branch to isolate the maintenance activity.
- If someone else has already checked out the directory, permission to check out the directory is denied. A separate shell window is displayed.

8 Cancel the checkout of the element if it is decided that no changes are to be made by typing into the directory input box of the **File Browser** the name of the directory to be checked in, press the **Enter** key, then follow the menu path **Version→Uncheckout→Unreserved: no comment**,

9 On the **File Browser** screen, follow the menu path **File→Exit**.

The ClearCase Graphical User Interface session is closed.

26.4.5 Importing files into ClearCase

Once the user has created a directory to place the science software files, ClearCase can be used to place a single file or multiple files in a UNIX directory structure under CM.

The following is a list of tools, and or assumptions:

1. A VOB and subdirectory are created to hold these files.
2. No object files or executables exist in the source code directory.

3. The PGE was received with a directory structure that contains various types of files.
4. These files will be entered into ClearCase and will maintain the same directory structure as the delivery structure.

26.4.5.1 Importing a Single File into ClearCase

Procedure:

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.
Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 At a UNIX prompt, type **cleartool setview *ViewName***, press **Enter**
The ***ViewName*** is the name of the ClearCase View.

4 At the UNIX prompt, type **cd *pathname***, then press the **Enter** key.
The ***pathname*** is the full path name of the subdirectory in the VOB into which the file is to be checked in.
If the desired directory cannot be seen, it could mean that the view has not been set or the properties of the view do not allow the directory to be seen; check with the CM Administrator.

5 At a UNIX prompt, type **cp *pathname/filename* .**, press **Enter** (note the space and then "dot" at the end of the command)**.**
The ***pathname*** is the full path name to the directory where the file to be checked in exists and ***filename*** is the file name of the file to be checked in.
This command copies a file over into the VOB area in preparation for checking it in.

6 At the UNIX prompt, type **cleartool checkout -nc .**, press **Enter** (note the space and then "dot" at the end of the command).
This command checks out the current directory (represented by the "dot") from ClearCase.
Adding a new file (or element) to a directory represents a modification of the directory. Hence, the directory must be checked out before a file can be checked in.

7 At a UNIX prompt, type **cleartool mkelem -nc *filename***, then press the **Enter** key.
The ***filename*** is the name of the file that was copied over in step 5 and is the file that will be checked into ClearCase.
This command creates a ClearCase element from the file in preparation for checking it in.
The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the make element step.

8 At the UNIX prompt, type **cleartool checkin -nc *filename***, then press the **Enter** key.
The ***filename*** is the name of the file to be checked into ClearCase.
This command performs the check in of the file.
The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the checkin step.

9 At the UNIX prompt, type **cleartool checkin -nc .**, press **Enter** (note the space and then "dot" at the end of the command).
This command checks in the current directory (represented by the "dot") into ClearCase.
The adding of an element (here, a file) represents a modification to the directory and hence, the new version of the directory must be checked back in.

The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the checkin step.

26.4.5.2 Importing Multiple Files into ClearCase

The DAP for the synthetic PGE contains only one source code module and a minimal number of other files. A real PGE will generally contain many source files, header files, and multiple other types of files stored in a standard type of directory structure which is retained when the PGE is packed into the tar file. The script provided by ClearCase is used for the purpose of making another load script to enter all of the DAP files along with the directory structure at one time. The final step of running the load script can only be performed by the DAAC Administrator.

The following procedure explains how to place the entire contents of a UNIX directory structure under ClearCase. A UNIX directory structure refers to all the files and subdirectories under some top-level directory.

This procedure is geared toward science software deliveries. In such cases, science software is delivered in the form of a UNIX *tar* files. A *tar* file has been unpacked (un*tar*-red) and the contents are to be placed under ClearCase configuration management.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

The following is a list of tools, and or assumptions:

1. A VOB and subdirectory are created to hold these files.
2. A ClearCase view is **not** required to perform this procedure.

Importing Multiple Files Into ClearCase

1 At a UNIX prompt, type **cd *ParentPathname***, then press the **Enter** key

The ***ParentPathname*** is the path name of the directory that *contains* the directory structure to be brought into ClearCase. This is *not* the VOB.

2 At the UNIX prompt, type **clearcvt_unix -r *DirName***, then press the **Enter** key.

The *DirName* is the name of the directory in which it and everything below it is to be brought into ClearCase.

A conversion script will be then be created. The -r causes all subdirectories to be recursively included in the script created.

3 Contact the VOB Administrator and request that the utility script cvt_script be run on the script created in step 2.

The VOB Administrator is the only one who can run the cvt_script because it modifies the VOB.

4 At this time the user logs out from this workstation. The VOB Administrator completes the procedure.

The remaining steps are accomplished by the VOB Administrator.

5 The VOB Administrator logs into the AIT Sun workstation by typing **username** then press the **Enter** key.

Cursor moves to the **Password** field.

6 Type the **password** then press the **Enter** key.

7 Invoke ClearCase by typing **xclearcase &** on the UNIX command line then press the **Enter** key.

The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
The ClearCase **View Tag Browser** screen is displayed listing available views.

8 To create a view for checking in the software change package, select a known View and press the **Enter** key. If you are using an existing view, select the desired existing view and proceed to step 14.

The File Browser window is displayed.

9 Select **File→Execute→Single Command.**

The String Browser window is displayed.

The prompt Enter shell command to run is displayed.

10 Invoke the make view command by typing **mkview [filename]** on the UNIX command line and press the **Enter** key.

The **tempdisp** window appears.

The **View [filename] Created Successfully** and the **Cache Updated for View [filename]** prompts are displayed.

11 Close the **tempdisp** window by clicking on the window and press the **Enter** key.

The **tempdisp** window closes.

12 Select the VOB where the software change package is to be imported then press the **Enter** key.

13 To create a subdirectory for the software change package in that VOB, which is a modification to the parent directory (for the VOB) the parent directory must be checked out by following the menu path **Version→Checkout→Reserved: no comment**.

In order to add new files to ClearCase, the directory in which the files are to be added must be checked out first.

ClearCase forces the checkout onto a maintenance branch to isolate the maintenance activity.

If someone else has already checked out the directory, permission to check out the directory is denied.

A separate shell window is displayed.

14 Start a shell process in a separate window by clicking on the shell icon button of the **File Browser** toolbar.

A separate shell window is displayed.

15 To run the script, type **cvt_script** then press the **Enter** key.

The VOB Administrator is the only person who can run the **cvt_script** because it modifies the VOB.

16 To check in the new directory, type into the directory input box of the **File Browser** screen: **path** [where **path** is the full path identification for the new directory (**directoryname**)], then press the **Enter** key. Then select **Versions→Checkin** from the menu.

17 To check in the parent directory (for the VOB), type into the directory input box of the **File Browser** screen: **VOBpath** (where **VOBpath** is the full path identification for the parent directory), then press the **Enter** key. Then select **Versions→Checkin** from the menu.

18 On the **File Browser** screen, follow menu path **File→Exit**.

The ClearCase Graphical User Interface session is closed.

26.4.6 Checking Out a File From ClearCase

If a configured file requires modification, then the file needs to be checked out of the configured directory and placed in a user directory. This will allow the file(s) to be modified.

The following is a list of tools, and or assumptions:

1. The file or directory must be an element created in ClearCase.
2. The view should be configured to ensure the correct version of the file or directory is seen.

Checking Out an Element/File from the Command Line

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt type **cleartool setview *ViewName***, then press the **Enter** key. The ***ViewName*** is the name if the user's view.

4 At a UNIX prompt type **cleartool checkout -nc *element*** then press the **Enter** key.

20. The ***element*** is the name of the file or directory that is to be checked out.
21. The **-nc** flag means "no comment" which will suppress the ClearCase prompting for a comment to be associated with the check out step.

Checking Out an Element/File from the File Screen Browser

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key. Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 Invoke ClearCase GUI by typing **xclearcase &** on the UNIX command line then press the **Enter** key.
The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
The ClearCase **View Tag Browser** screen is displayed listing available views.

4 To check out the directory where the controlled files were place, type into the directory input box of the **File Browser** screen: **path** [where **path** is the full path identification for the directory (**directoryname**)], then press the **Enter** key. Then select **Versions→Checkout** from the menu.

5 Select **File→Execute→Single Command.**
The String Browser window is displayed.
The prompt **Enter shell command to run** is displayed.

6 To determine editing privileges, type **ls -l,** then press the **Enter** key.
A prompt displaying read/write/execute privileges will be displayed. There will be three groupings:
- **User Group Others**
- **r=read, w=write, x=execute**

7 If you have editing/execute privileges, you can revise the contents of the file with any text editor.

8 To checkin a controlled file, select **Versions→Checkin** from the menu.
The file/directory will be checked in to ClearCase and the version will be updated.

26.4.7 Checking a Modified Element into ClearCase

This procedure explains how to check in a modified element to ClearCase. An element refers to a directory or file in ClearCase, that is, under configuration management. Modifications made to a file or directory cannot be saved in ClearCase unless the file or directory had been checked out first.

The following is a list of tools, and or assumptions:

1. A VOB exits and is mounted at a known UNIX directory.
2. A ClearCase view exists for the SSI&T operator.
3. The element or file has been checked out and modified.
4. The modified file is now in the user's directory on the VOB from which it was checked out.

26.4.7.1 Checking a Modified Element/File into ClearCase

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 At a UNIX prompt, type **cleartool setview *ViewName***, then press the **Enter** key.

The ***ViewName*** is the name of the user's view.

4 At the UNIX prompt, type **cleartool checkin -nc *filename***, then press the **Enter** key.

The ***filename*** is the name of the file (full path name allowed) that is to be checked out (and later modified).

The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check out step.

This command checks in the current directory.

5 This step is optional; it is performed when ClearCase does not accept a checkin because the element was not modified. In this case, the check out must be canceled. At a UNIX prompt, type **cleartool uncheckout -nc *filename***, then press the **Enter** key.

The ***filename*** is the name of the file or directory (full path name allowed) checked out.

This command cancels the check out of an element/file.

26.5 Standards Checking of Science Software

The purpose of standards checking is to verify that the source files of the science software are compliant with the ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document.

26.5.1 Checking FORTRAN 77 ESDIS Standards Compliance

The ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document requires all FORTRAN 77 code to be compliant with the ANSI FORTRAN 77. The COTS used for this task is FORCHECK.

The following is a list of tools, and or assumptions:

Assumptions:

1. The FORTRAN 77 science software source code is available, accessible, and has read permissions for the user.
2. SSIT Manager is available for use.

FORCHECK is available only on the AIT Suns.

To check for ESDIS standards compliance in FORTRAN 77 code, execute the procedure steps that follow:

---

1 If not already on an AIT Sun, log into one from your machine.

22. Once logged onto proper Sun, remember to set the DISPLAY environmental variable to point to your X Window screen.

2 If required, at the UNIX prompt on the AIT Sun, type **cleartool setview *ViewName***, press **Return**.

23. The ***ViewName*** is the name of a view allowing the FORTRAN 77 source files to be accessible.
24. This step is only necessary if any of the FORTRAN 77 source files are in ClearCase (in the VOB under configuration management).

3 If your general environment setup does not include transparent access to the SSIT Manager GUI, then you need to set that up. One way to do it is as follows:

25. Set up an alias, manually or from shell script, to set up preliminary environment. At UNIX prompt, type **alias do_buildrc "source /usr/ecs/TS1/CUSTOM/bin/DPS/.buildrc"**
26. Set up an alias, manually or through shell script, to invoke SSIT Manager. At UNIX prompt, type **alias do_ssit_man "/usr/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgr ConfigFile /usr/ecs/TS1/CUSTOM/cfg/EcDpAtMG.CFG ecs_mode TS1& "**

4 Set up the preliminary environment (do_buildrc). This only needs to be done once per session. Then, run SSIT Manager (do_ssit_man).

27. Type **do_buildrc**
28. Type **do_ssit_man**

5 Once the SSIT Manager comes up, the following steps need to be taken to invoke FORCHECK

29. From the top menu bar, select **Tools.**
30. From the Tools menu, select **Standards Checkers.**
31. From the Standards Checkers menu, select **FORCHECK.**
32. See Figure 26.8.5-2. for a screen snapshot of this step.

6 A separate FORCHECK window will now open.

33. The user will be prompted for input. The first prompt will be *global option(s) and list file?*
34. The second prompt will be *local option(s) and file(s)?*
35. The second prompt will be repeated until there is a blank line and carriage return.
36. In order to understand what the proper responses should be, the user is encouraged to find hardcopy documentation for FORCHECK or to use the UNIX man facility and type *man forchk* .

7 At the UNIX prompt on the AIT Sun, type **vi *FORCHECKoutput***, press **Return**.

37. The ***FORCHECKoutput*** is the file name for the output file produced in step 6.
38. The ***FORCHECKoutput*** file will contain any warnings, errors, and other messages from FORCHECK. A summary will be at the bottom of the file.
39. Any text editor may be used for this procedure step.

8 At the UNIX prompt on the AIT Sun, type **vi *ListFile***, press **Return**.

40. The ***ListFile*** is the file name for the list file specified at the FORCHECK prompt.

41. The ***ListFile*** file will contain FORCHECK messages similar to the ***FORCHECKoutput*** file embedded in the source code listing.

Any text editor may be used for this procedure step.

26.5.2 Checking for ESDIS Standards Compliance in Fortran 90

This procedure describes how to use the Fortran 90 compiler flags on the SPR SGI machines to check science software written in Fortran 90 for ESDIS standards compliance.

Unlike with FORTRAN 77, no COTS tool is used to check Fortran 90 science software. Instead, this procedure describes how to use the compiler to perform the checking (ESDIS standards for Fortran 90 are ANSI). Since the Fortran 90 compiler is used, the checking for standards compliance can be naturally tied in with building the science software (since this procedure will produce object files suitable for linking). However, in this procedure, the building of the software (compiling *and* linking) is deferred to a later procedure.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The Fortran 90 science software source code is available, accessible, and has read permissions for the user.
2. Required Status Message Facility (SMF) files have been compiled.
3. The C shell (or a derivative) is the current command shell.
4. The Fortran 90 compiler is available on the SPR SGI.

To check for ESDIS standards compliance in Fortran 90 code, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SPR SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SPR SGI.

2 At the UNIX prompt on the SPR SGI, set up the proper environment for the compiler to be used by typing **source *ToolkitPathname* /bin/*sgiXX*/pgs-dev-env.csh .**

- ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version .
- The directory ***sgiXX*** should be replaced with **sgi32** or **sgi64** as appropriate for the specific compiler desired.
- For example, on the mini-DAAC platform p0spg01, type **source /data3/ecs/TS1/CUSTOM/daac_toolkit_f90/TOOLKIT/bin/sgi64/pgs-dev-env.csh .** This will set up the various environment parameters, such as PGSHOME, to enable the 64 bit version of the FORTRAN 90 compiler to be run.

3 If required, at the UNIX prompt on the SPR SGI, type **cleartool setview *ViewName***, press **Return**.
- The ***ViewName*** is the name of a view allowing the Fortran 90 source files to be accessible.
- This step is only necessary if any of the Fortran 90 source files are in ClearCase (in the VOB under configuration management).

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.
- The ***SrcPathname*** is the full path name to the location of the Fortran 90 source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the Fortran 90 source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **f90 -c -ansi *[*-I$PGSINC*] [*-I$HDFINC*] [[*-IOtherIncFiles*…] SourceFiles* >& *ReportFile***, press **Return**.
- The terms in square brackets (*[ ]*) are used to optionally specify locations of include and module (.mod) files. The **$PGSINC** already contains the SDP Toolkit include directory and **$HDFINC** already contains the HDF include directory. The **OtherIncFiles** represents one or more additional include or module directories.
- The ***SourceFiles*** is a list (space delimited) of Fortran 90 source files or a wildcard template (*e.g.* *.f90).
- The **>&** is a C shell construct that causes standard error (where the output from the Fortran 90 compiler normally emerges) to be redirected to a file.
- The ***ReportFile*** is the file name under which to save the results of the compile process.
- The **-c** flag causes only compilation (no linking).
- The **-ansi** flag enables ANSI checking.
- Apply the terms in square brackets only as necessary. Do not include the brackets in the actual command. See example below.
- Do not use the **-I** option for include or module files that are in the standard directories or in the current directory.
- The makefile for the science software may contain the names of additional include files needed by the software.
- For example, type **f90 -c -I$PGSINC -I$HDFINC -I/ecs/modis/pge5/include/ *.f90 >& pge10.report**, press **Return**.

6 At the UNIX prompt on the SPR SGI, type **vi *ReportFile***, press **Return**.
- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step.

26.5.3 Checking for ESDIS Standards Compliance in C

This procedure describes how to use the C compiler flags on the SPR SGI machines to check science software written in C for ESDIS standards compliance.

Unlike with FORTRAN 77, no COTS tool is used to check C science software. Instead, this procedure describes how to use the compiler to perform the checking (ESDIS standards for C are

essentially ANSI). Since the C compiler is used, the checking for standards compliance can be naturally tied in with building the science software (since this procedure will produce object files suitable for linking). However, in this procedure, the building of the software (compiling *and* linking) is deferred to a later procedure.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The C science software source code is available, accessible, and has read permissions for the user.
2. Required Status Message Facility (SMF) files have been compiled.
3. The C shell (or a derivative) is the current command shell.

- The C compiler is available on the SPR SGI.

To check for ESDIS standards compliance in C code, execute the procedure steps that follow:

---

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SPR SGI.
Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SPR SGI.

2 At the UNIX prompt on the SPR SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.
The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version.
The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

3 If required, at the UNIX prompt on the SPR SGI, type **cleartool setview *ViewName***, press **Return**.
The ***ViewName*** is the name of a view allowing the C source files to be accessible.
This step is only necessary if any of the C source files are in ClearCase (in the VOB under configuration management).

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.
The ***SrcPathname*** is the full path name to the location of the C source files to be checked.
The ***SrcPathname*** will be in the ClearCase VOB is the C source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **cc -c -ansi *[*-I$PGSINC*] [*-I$HDFINC*] [[*-IOtherIncFiles*…] SourceFiles* >& *ReportFile***, press **Return**.
The terms in square brackets (*[ ]*) are used to optionally specify locations of include and module (.mod) files. The **$PGSINC** already contains the SDP Toolkit include directory and **$HDFINC** already contains the HDF include directory. The **OtherIncFiles** represents one or more additional include directories.
The ***SourceFiles*** is a list (space delimited) of C source files or a wildcard template (*e.g.* *.c).

The **>&** is a C shell construct that causes standard error (where the output from the C compiler normally emerges) to be redirected to a file.
The ***ReportFile*** is the file name under which to save the results of the compile process.
The **-c** flag causes only compilation (no linking).
The **-ansi** flag enables ANSI checking.
Apply the terms in square brackets only as necessary. Do not include the brackets in the actual command. See example below.
Do not use the **-I** option for include files that are in the standard directories (*e.g.* /usr/include) or in the current directory.
The makefile for the science software may contain the names of additional include files needed by the software.
For example, type **cc -c –ansi-I$PGSINC -I$HDFINC -I/ecs/modis/pge5/include/ *.cÄ >& pge10.report**, press **Return**.

6 At the UNIX prompt on the SPR SGI, type **vi *ReportFile***, press **Return**.
The ***ReportFile*** is the file name for the compilation results as produced in step 5.
Any text editor may be used for this procedure step.

26.5.4 Checking for ESDIS Standards Compliance in Ada

This procedures describes how to use Ada compilers on the SPR SGI machines to check science software written in Ada for ESDIS standards compliance.

Unlike with FORTRAN 77, Fortran 90, or C, Ada compilers are subjected to a validation process by the DoD Ada Committee. Thus, any code that compiles successfully by a validated compiler is, by definition, fully ANSI compliant. Since the Ada compiler is used, the checking for standards compliance can be naturally tied in with building the science software (since this procedure will produce object files suitable for linking). However, in this procedure, the building of the software (compiling *and* linking) is deferred to a later procedure.

26.5.4.1 Checking for ESDIS Standards Compliance in Ada: Verdix COTS

This procedure describes compiling Ada software using the COTS Verdix Ada Development System (VADS) which provides a complete environment for building (and developing) Ada software. See the *gcc* compiler in compiling Ada code.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The Ada science software source code is available, accessible, and has read permissions for the user.
2. The C shell (or a derivative) is the current command shell.

- The Ada compiler is available on the SPR SGI.

To check for ESDIS standards compliance in Ada code, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SPR SGI.
- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SPR SGI.

2 If required, at the UNIX prompt on the SPR SGI, type **cleartool setview *ViewName***, press **Return**.
- The ***ViewName*** is the name of a view allowing the Ada source files to be accessible.
- This step is only necessary if any of the Ada source files are in ClearCase (in the VOB under configuration management).

3 At the UNIX prompt on the SPR SGI, type **setenv SGI_ABI -32**, press **Return**.
- This command sets the environment variable **SGI_ABI** for 32-bit mode compilation.

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.
- The ***SrcPathname*** is the full path name to the location of the Ada source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB if the Ada source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **a.mklib**, press **Return**.
- This command creates a VADS library directory. All Ada compilation must occur in a VADS Ada library.

6 At the UNIX prompt on the SPR SGI, type **a.make -v -f *SourceFiles* >& *ReportFile***, press **Return**.
- The ***SourceFiles*** is a list (space delimited) of Ada source files or a wildcard template (*e.g.* *.ada).
- The **>&** is a C shell construct that causes standard error (where the output from the Ada compiler normally emerges) to be redirected to a file.
- The ***ReportFile*** is the file name under which to save the results of the compile process.
- The **-v** flag enables verbose output.
- The **-f** flag indicates that what immediately follows are the source files. The order of the flags is therefore important.

7 At the UNIX prompt on the AIT Sun, type **vi *ReportFile***, press **Return**.
- The ***ReportFile*** is the file name for the compilation results as produced in step 6.
- Any text editor may be used for this procedure step.

26.5.4.2 Checking for ESDIS Standards Compliance in Ada: GNU *gcc* Compiler

This procedure describes compiling Ada software using the GNU C compiler, *gcc*.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The Ada science software source code is available, accessible, and has read permissions for the user.
2. The C shell (or a derivative) is the current command shell.

The GNU *gcc* compiler is available on the SPR SGI.

To check for ESDIS standards compliance in Ada code, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SPR SGI.
- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SPR SGI.
- It is recommended that this procedure begin within a new command shell on the SPR SGI.

2 If required, at the UNIX prompt on the SPR SGI, type **cleartool setview *ViewName***, press **Return**.
- The ***ViewName*** is the name of a view allowing the Ada source files to be accessible.
- This step is only necessary if any of the Ada source files are in ClearCase (in the VOB under configuration management).

3 At the UNIX prompt on the SPR SGI, type **setenv SGI_ABI -32**, press **Return**.
- This command sets the environment variable **SGI_ABI** for 32-bit mode compilation.

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.
- The ***SrcPathname*** is the full path name to the location of the Ada source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the Ada source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **gcc -c -gnat83 *SourceFiles* >& *ReportFile***, press **Return**.
- The ***SourceFiles*** is a list (space delimited) of Ada source files or a wildcard template (*e.g.* *.ada).
- The **>&** is a C shell construct that causes standard error (where the output from the *gcc* compiler normally emerges) to be redirected to a file.
- The ***ReportFile*** is the file name under which to save the results of the compile process.
- The **-c** flag causes only compilation (no linking).
- The **-gnat83** enables compilation of Ada using the 1983 Ada Standard. Note that without this flag, the compiler would assume the 1995 Ada proposed Standard.

6 At the UNIX prompt on the SPR SGI, type **vi *ReportFile***, press **Return**.
- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step

26.5.5 Prohibited Function Checker

The use of certain functions in the PGE is prohibited. The Prohibited Function Checker (Figure 26.5.4-1) is used to check C, FORTRAN 77, FORTRAN 90, and Ada language source files for the occurrence of functions that are prohibited in the ECS DAAC production environment.

26.5.5.1 Checking for Prohibited Functions: Command-Line Version

This procedure describes using the command-line version of the Prohibited Function Checker to check science software for the prohibited functions.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The source files to be checked are available, accessible, and have read permissions for the operator.
2. Source files to be checked are Ada, C, FORTRAN 77, Fortran 90, C shell, Korn shell, Bourne shell, or Perl and have recognizable file name extensions.

To check for prohibited functions in delivered source files, execute the procedure steps that follow:

---

1 If required, at the UNIX prompt on an AIT Sun, type **cleartool setview *ViewName***, press **Return**.
- The ***ViewName*** is the name of a view allowing the source files to be accessible.
- This step is only necessary if any of the source files are in ClearCase (in the VOB under configuration management).

2 At the UNIX prompt on the AIT Sun, type **cd *SrcPathname***, press **Return**.
- The ***SrcPathname*** is the full path name to the location of the source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB if the source files are checked into ClearCase.
- The ***SrcPathname*** can contain other directories that contain source files and/or more directories. The Prohibited Function Checker will search out all source files in subdirectories recursively.

3 At the UNIX prompt on the AIT Sun, type **/data3/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgrBadFunc ConfigFile /data3/ecs/TS1/CUSTOM/cfg/EcDpAtBA.CFG *FilesOrDirectories* > *ResultsFile***, press **Return**.
- The ***FilesOrDirectories*** is a list of source file names or directory names of directories containing source files.
- The ***ResultsFile*** is the file name for the results that are output.
- For example, type **/data3/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgrBadFunc ConfigFile /data3/ecs/TS1/CUSTOM/cfg/EcDpAtBA.CFG main.c utils/ > myOutput**, press **Return**. Here, **main.c** is a source file and **utils/** is a directory that contains other source files.

4 At the UNIX prompt on the AIT Sun, type **vi *ResultsFile***, press **Return**.
- The ***ResultsFile*** is the file name for the output results as produced in step 3.
- Any text editor may be used for this procedure step.

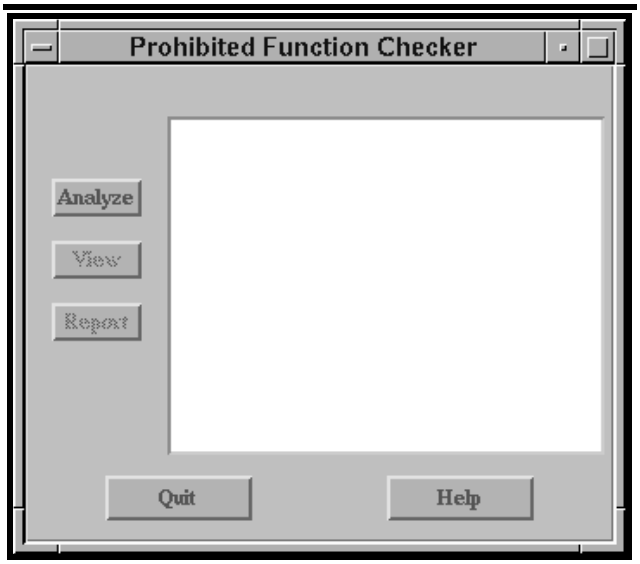


Figure 26.5.5-1. Prohibited Function Checker

Prohibited Function Checker GUI

1 From the SSIT Manager, select **Tools** → **Standards Checkers** → **Prohibited Function Checker** from the menu.

The Prohibited Function Checker GUI will be displayed.

2 In the Prohibited Function Checker GUI, click on the **Analyze** button.

The File Selector GUI will be displayed.

3 Within the **Directories** subwindow, double click on the desired directory.

Repeat this step until the directory with the source files to be checked are displayed in the **Files** subwindow.

4 Within the **Files** subwindow, click on the source files to be checked. Each file clicked on will be highlighted.

To choose groups of contiguous files, hold down the left mouse button and drag the mouse.

To choose non-contiguous files, hold down the Control key while clicking on file names.

5 In the File Selector GUI, click on the **OK** button.

The File Selector GUI will disappear.

The files selected in step 5 will be displayed in the Prohibited Function Checker GUI window as they are being checked.

6 In the Prohibited Function Checker GUI, click on the **Report** button.

The **Report** GUI will be displayed.

For each file, a list of prohibited functions found will be displayed.

7 Optionally, click on the **Print** button or the **Save** button.

Choose **Save** to save the results to a file; choose **Print** to have the results printed on the default printer.

Choosing **Save** will bring up a GUI labeled **Save To File**. Specify the directory and file name in which to save the results file.

8 Optionally, in the Prohibited Function Checker GUI, highlight one of the source files listed. Then click on **View**.

The **Source Code** GUI will be displayed.

Occurrences of prohibited functions found in that source file will be highlighted.

Click on the **Next** button to bring into the window successive occurrences of prohibited functions (the **Next** button does not bring in the next source file).

Click on the **Done** button to close the **Source Code** GUI. Other source files may be examined similarly, one at a time.

9 In the Prohibited Function Checker GUI, click on the **Quit** button.

The Prohibited Function Checker GUI will disappear.

This ends the session.

Figure 26.5.5-2. Invoking the Prohibited Function Checker

26.5.6 Checking Process Control Files

The next task to accomplish is to check that the PCFs are syntactically correct and contain all necessary information for PGEs to run within the ECS DAAC production environment. Only one PCF can be associated with a PGE. The following procedure describes how to check PCFs for valid syntax and format, both using the GUI and the command line interface.

26.5.6.1 Checking Process Control Files GUI

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. The Process Control File(s) are available, accessible, and have read permissions.

If the source code files to be checked are in a VOB in ClearCase, a view has been set before the SSIT Manager was started.

Checking Process Control Files GUI

1 From the SSIT Manager, select **Tools** → **Standards Checkers** → **Process Control File Checker** from the menu, see figure 26.5.6-1.

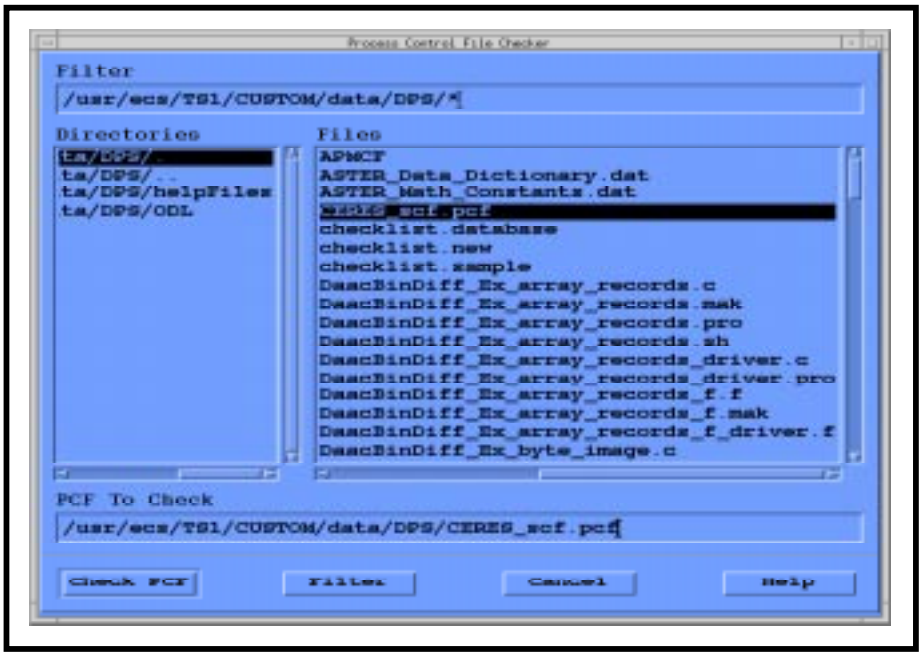


Figure 26.5.6-1. Process Control File Checker GUI

The Process Control File Checker GUI will be displayed.

2 In the **Directories** subwindow, double click on the desired directory.

Repeat this step until the directory with the PCF(s) to be checked are displayed in the Files window.

Use the **Filter** subwindow to limit which files are displayed.

3 Within the **Files** subwindow, click on the PCF to be checked**.**

The file clicked on will be highlighted.

Only one PCF can be checked at a time.

4 Click on the **Check PCF** button.

A GUI labeled **PCF Checker Results** will be displayed.

Results will be displayed in this window.

5 Optionally, click on the **Save** button or on the **Print** button.

Choose **Save** to save the results to a file; choose **Print** to have the results printed on the default printer.

Choosing **Save** will bring up a GUI labeled **Save To File**. Specify the directory and file name in which to save the results file.

Choosing **Print** and then clicking on the **OK** button will send the results to the default printer.

6 Click on the **Check Another** button or on the **Quit** button.

Choosing **Check Another** allows another PCF to be checked. Repeat steps 2 through 5.

Choosing **Quit** causes the Process Control File Checker GUI to disappear and ends the session.

26.5.6.2 Checking Process Control Files: Command-Line Version

This procedure describes using the command-line version of the Process Control File Checker to check process control files delivered with the science software.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The PCF files to be checked are available, accessible, and have read permissions for the operator.
2. You will need the command **pccheck.sh**. One way to see if this is available is to type **which pccheck.sh**, press **Return**. If a path is displayed, then the directory is in your path. On the mini-DAAC Sun platform **p0ais01,** the pathname for the command is /ecs/formal/TOOLKIT/bin/sun5/pccheck.sh . In this case, you will have to set a ClearCase view to access that area.

To check Process Control Files, execute the procedure steps that follow:

1 If required, at the UNIX prompt on an AIT Sun, type **cleartool setview *ViewName***, press **Return**.

- The ***ViewName*** is the name of a view allowing the Process Control File(s) to be accessible.
- This step is only necessary if any of the Process Control Files are in ClearCase (in the VOB under configuration management).

2 At the UNIX prompt on AIT Sun, type **cd *PCFpathname***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the Process Control File(s) to be checked.
- The ***PCFpathname*** will be in the ClearCase VOB if the Process Control Files are checked into ClearCase.

3 At the UNIX prompt on an AIT Sun, type **/ecs/formal/TOOLKIT/bin/sun5/pccheck.sh -i *PCFfilename* > *ResultsFile***, press **Return**.

- The ***PCFfilename*** is the full path name (directory and file name) to the Process Control File to check.
- The ***ResultsFile*** is the file name for the results that are output.
- The PCF Checker is also available on the SPR SGI machines. The easiest way to access it is to set a SDP Toolkit environment (any will do for purposes here, see Section 9.2) and type **$PGSBIN/pccheck.sh -i *PCFfilename* > *ResultsFile***, press **Return**.

4 At the UNIX prompt on the SPR SGI, type **vi *ResultsFile***, press **Return**.

- The ***ResultsFile*** is the file name for the output results as produced in step 4.
- Any text editor may be used for this procedure step.

26.5.7 Extracting Prologs

The Project standards and guidelines are contained in the latest version of the document *Data Production Software and Science Computing Facility (SCF) Standards and Guidelines* (423-16-01). This ESDIS document mandates that science software delivered to the DAACs to be integrated into the ECS contain prologs in the source files. Prologs are internal documentation containing information about the software. The details are specified in the ESDIS document. Prologs must be at the top of every function, subroutine, procedure, or program module.

This procedure describes using the Prolog Extractor to extract prologs into a file. Note that the prolog extractor only extract the prologs it finds. It does not check the contents of prologs.

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. Prologs are assumed to be delimited by particular delimiters depending on the language type. Delimiters are listed in the table below:

Prolog Delimiters

Language	Type	Delimiter
FORTRAN 77	source	!F77
Fortran 90	source	!F90
C	source	!C
Ada	source	!Ada
FORTRAN 77	include	!F77-INC
Fortran 90	include	!F90-INC
C	include	!C-INC
Any Language	any	!PROLOG
All Languages	The end delimiter is always !END	

The Prolog Extractor recognizes the language type of the file by its file name extension. The table below lists assumed file name extensions:

File Name Extensions

File Type	File Name Extensions
FORTRAN 77	f, f77, ftn, for, F, F77, FTN, FOR
Fortran 90	f90, F90, f, F
FORTRAN 77/Fortran 90 include	inc, INC
C	c
C/C++ header	h
Ada	a, ada

26.5.7.1 Extracting Prologs

The Prolog Extractor can be started from the UNIX prompt. To do this, at the UNIX prompt on the AIT Sun, type **/data3/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgrPrologs**, press **Return**

or

1 From the SSIT Manager, select the **Tools → Standards Checkers → Prolog Extractor** from the menu.

An xterm will be displayed on the AIT Sun.

Select the default ConfigFile. The output goes to a file called Prologs.txt in the directory from which the SSIT Manager was started.

The Prologs.txt file can be viewed by changing directories to the SSIT Manager directory and invoking a text editor. The file may also be sent to a printer.

2 At the **Files(S)? (-h help)** prompt, type in the file names and/or directory names containing the files.

Separate items with spaces.

The contents of the directory will be search recursively for files with valid file name extensions.

Use **./** to indicate current directory.

The time needed for the Prolog Extractor could be very long for large numbers of files and directories.

When extraction is complete, the message **Output written to file: ./prologs.txt** will be displayed.

3 At the program prompt **Hit Enter for another, "q <Enter>" to quit:** , press **Enter** to repeat process with another set of source files or type **q** and press **Enter** to quit.

- The xterm will disappear.

4 At a UNIX prompt on the AIT Sun, type **vi prologs.txt,** then press the **Enter** key.

The extracted prologs file, named **prologs.txt,** will be brought into the editor.

The default location of the **prologs.txt** file is the directory from which the SSIT Manager was invoked.

5 Once the extracted prologs file has been examined, exit the editor.

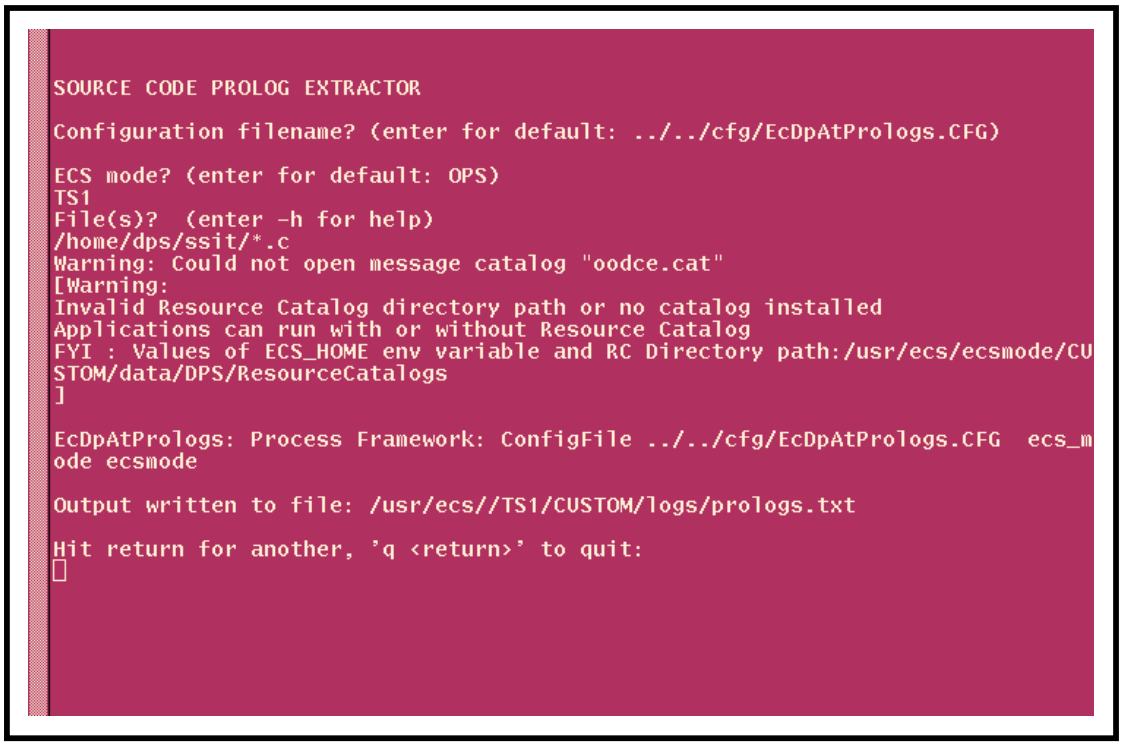


Figure 26.5.7-1. Prolog Extractor Sample Run.

26.6 Compiling and Linking Science Software

Science software to the DAACs is in the form of source files. In order to be run and tested within the ECS, this science software has to be compiled and linked to form the binary executables that run within the PGEs. Science software is developed at independent Science Computing Facilities (SCFs) using the SDP Toolkit. The SDP Toolkit allows science software to be developed for ECS at independent SCFs. Once delivered to the DAACs for SSI&T, science software needs to be compiled and linked to one of the SDP Toolkit versions resident at the DAAC. The (PCFs) Process Control Files provide the interface between the science software and the production system in the ECS. Since the process control files delivered to the DAACs for SSI&T were created and used at the SCFs, the path names in the PCF will need to be checked and revised to work at the DAACs.

To save time for the SSI&T Training Lesson, the compile and link with the SCF Version of the Toolkit will be omitted. The procedures are included in the student guide for future reference.

The next step is to set up a DAAC version SDP Toolkit environment, compile the PGE, and link to the DAAC Toolkit. This procedure will be performed at the SSI&T Training. The procedure steps for the two processes are the same except for the set up for the Toolkit environment and link with the corresponding Toolkit library.

26.6.1 Updating the Process Control File

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. A PCF for the PGE has been delivered and is available, accessible, and has read permissions.

To update the PCF, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SGI.

2 If required, at the UNIX prompt on the SGI, type **cleartool setview *ViewName***, press **Return**.

- The ***ViewName*** is the name of a view allowing the PCF to be accessible.
- This step is only necessary if the PCF is in ClearCase (in the VOB under configuration management).

3 At the UNIX prompt on the Sun or on the SGI, type **cd *PCFpathname***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the PCF. This location will be in the ClearCase VOB if the PCF is under configuration management.

4 At the UNIX prompt on the Sun or on the SGI, type **cleartool checkout -nc *PCFfilename***, press **Return**.

- The ***PCFfilename*** is the file name of the PCF that is to be checked out (and later modified). The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check out step.

5 Run the Process Control File Checker on the delivered PCF.

- This will verify that the delivered PCF is correct before editing.

6 At a UNIX prompt on the Sun, type **vi *PCFfilename***, press **Return**.

- The ***PCFfilename*** is the file name of the PCF to update.
- Any text editor may be used such as *emacs*. For example, **emacs AST02.pcf**, press **Return**.

7 In the file, make changes to the default directories specified in each section of the PCF. All path names specified in the PCF must exist on the SGI.

- Each section begins with a line consisting of a **?** in the first column followed by a label:

```
? PRODUCT INPUT FILES
? PRODUCT OUTPUT FILES
? SUPPORT INPUT FILES
? SUPPORT OUTPUT FILES
? INTERMEDIATE INPUT
? INTERMEDIATE OUTPUT
? TEMPORARY I/O
```

- Each of the above section heading lines will then be followed (not necessarily immediately; there may be comment lines) by a line that begins with a **!** in the first column. These lines specify the default path names for each section.
  - If the line reads:
    ```
    ! ~/runtime
    ```
    leave it unchanged. The tilde (~) is a symbol that represents $PGSHOME.
  - If another path name is listed instead, it will probably need to be changed to a path name that exists at the DAAC on the SGI. When specifying a path name, use an absolute path name, not a relative path name.

8 In the file, look for science software specific entries in each section and make changes to the path names (field 3) as necessary. All path names specified in the PCF must exist on the SGI.

- The science software specific entries will have logical IDs (first field) *outside* of the range 10,000 to 10,999.
- Where necessary, replace the path names in the third field of each entry with the path names appropriate to the DAAC environment.
- Do not alter file entries that are used by the SDP Toolkit itself. These have logical IDs *in* the range 10,000 to 10,999.
- For example, if the following entry was found in the PCF:
  ```
  100|A.granule|/MODIS/run/input||||1
  ```
  change /MODIS/run/input to the appropriate path name in the DAAC where the file A.granule is stored.
- When specifying a path name, use an absolute path name, not a relative path name.
- Do not include the file name with the path name. The file name belongs in field 2 by itself.

9 In the file, verify that the SUPPORT OUTPUT FILES section contains an entry to the shared memory pointer file.

- Look for the entry:
  ```
  10111|ShmMem|~/runtime||||1
  ```
  The third field may be blank; this will work too.
- If this entry is not within this section, add it.

10 Once changes have been made to the PCF, save the changes and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.
- For other editors, refer to that editor's documentation.

11 Again, run the Process Control File Checker on the PCF.

12 If the PCF had been checked out of ClearCase, at the UNIX prompt on the SGI, type **cleartool checkin -nc *PCFfilename***, press **Return**.

- The ***PCFfilename*** is the file name of the modified PCF. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check in step.

26.6.2 Setting up a SDP Toolkit Environment

The purpose of the SDP Toolkit is to allow science software to be developed for ECS at independent SCFs and to provide:

- An interface to the ECS system, including PDPS and CSMS and information management.
- A method for Science software to be portable to different platforms at the DAAC.
- A method to reduces redundant coding at the SCF.
- Value added functionality for science software development.

The SDP Toolkit is divided into two groups of tools:

26.6.2.1 Mandatory Tools

- Error and Status Message Facility (SMF) - provides general error handling, status log messaging, and interface to CSMS services.
- Process Control Tools - provides the primary interface to the PDPS. Allows access to physical filenames and file attributes and retrieval of user defined parameters.
- Generic Input/Output - provides the means to open and close support, temporary and intermediate duration files.
- Memory Allocation Tools - simple wrappers on native C functions which track memory usage in the SDPS, and shared memory tools which enable the sharing of memory among executables within a PGE.

26.6.2.2 Optional Tools

- Ancillary Data Access - provides access to NMC data and Digital Elevation (DEM) data.
- Celestial Body Position - locates the sun, moon and the planets.
- Coordinate System Conversion - coordinate conversions between celestial reference.
- Constant and Unit Conversion - physical constants and unit conversions.
- IMSL - mathematical and statistical support.

In the description of the Toolkit routines, descriptive information is presented in the following format:

TOOL TITLE

NAME:	Procedure or routine name
SYNOPSIS:	C: C language call
FORTRAN:	FORTRAN77 or Fortran90 language call
DESCRIPTION:	Cursory description of routine usage

INPUTS:	List and description of data files and parameters input to the routine
OUTPUTS:	List and description of data files and parameters output from the routine
ENTERS:	List of returned parameters indicating success, failure, etc.
EXAMPLES:	Example usage of routine
NOTES:	Detailed information about usage and assumptions
REQUIREMENTS:	Requirements from PGS Toolkit Specification, Oct. 93 which the routine satisfies

The science software delivered to the DAACs is expected to work with either the SCF SDP Toolkit or the DAAC SDP Toolkit which are both installed each DAAC. During the pre-SSI&T initial testing, the SCF Toolkit should be used.

There are several versions of the SCF/DAAC SDP Toolkit installed on the SGI Power Challenges at the DAACs for the Release 4 system. The toolkit versions at the DAACs differ according to:

Object Type - The operating system on the SGI Power Challenges on Release 4 is IRIX 6.2, a 64-bit operating system. To be backward compatible, the SGI operating system will allow new 64-bit and 32-bit objects to be built as well as the older 32-bit machines. Each of these object types are designated by placing a cc flag on the command line to enable a particular mode with the SGI C compiler.

New 64-bit: cc flag = -64
New 32-bit: cc flag = -n32
Old 32-bit: cc flag = -32 (SCF's only)

Library Type - The SDP Toolkit uses different libraries depending upon whether FORTRAN 77 or FORTRAN 90 source code is being linked. If C source code is to be linked, then either language version of the library will work.

The following Table summarizes the available SDP Toolkits used by the SGI science processors.

Table 26.6.2-1. SDP Toolkits used by the SGI science processors.

SDP Version	Language Type	Library Object Type	$PGSHOME	$PGSBIN
SCF	FORTRAN 77 or C	Old 32-bit mode	$CUSTOM/TOOLKIT/toolkit /bin/sgi_scf_f77/	$ CUSTOM/TOOLKIT/toolkit /bin/sgi_scf_f77/
SCF	Fortran 90 or C	Old 32-bit mode	$ CUSTOM/TOOLKIT/toolkit /bin/sgi_scf_f90/	$ CUSTOM/TOOLKIT/toolkit /bin/sgi_scf_f90/
SCF	FORTRAN 77 or C	New 32-bit mode	$ CUSTOM/TOOLKIT/toolkit /bin/sgi32_scf_f77/	$ CUSTOM/TOOLKIT/toolkit /bin/sgi32_scf_f77/
SCF	Fortran 90 or C	New 32-bit mode	$ CUSTOM/TOOLKIT/toolkit /bin/sgi32_scf_f90/	$ CUSTOM/TOOLKIT/toolkit /bin/sgi32_scf_f90/
SCF	FORTRAN 77 or C	64-bit mode	$ CUSTOM/TOOLKIT/toolkit /bin/sgi64_scf_f77/	$ CUSTOM/TOOLKIT/toolkit /bin/sgi64_scf_f77/
SCF	Fortran 90 or C	64-bit mode	$CUSTOM/TOOLKIT/toolkit /bin/sgi64_scf_f90/	$CUSTOM/TOOLKIT/toolkit /bin/sgi64_scf_f90/
DAAC	FORTRAN 77 or C	64-bit mode	$CUSTOM/TOOLKIT/toolkit /bin/sgi64_daac_f77/	$CUSTOM/TOOLKIT/toolkit /bin/sgi64_daacf77/ /
DAAC	Fortran 90 or C	64-bit mode	$ CUSTOM/TOOLKIT/toolkit /bin/sgi64_daac_f90/	$CUSTOM/TOOLKIT/toolkit /bin/sgi64_daac_f90/

26.6.2.3 Setting Up the SDP Toolkit Environment

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The C shell (or a derivative) is the current command shell.

To check set up a SDP Toolkit environment, execute the procedure steps that follow:

1 At the UNIX prompt on the SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**.

- The ***ToolkitPathname*** is the home directory of the particular SDP Toolkit version being used. Refer to Table 26.6.2-1. Note that the setting of PGSHOME shown in this table may differ in your local DAAC.
- Korn shell users, type **PGSHOME=*ToolkitPathname*; export PGSHOME**, press **Return**.

2 At the UNIX prompt on the SGI, type **source $PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***sgiX*** is one of: **sgi** for 32-bit version of the Toolkit or **sgi64** for 64-bit version of the Toolkit. Refer to the last column of Table 26.6.2-1. for path names to the file to source.
- Korn shell users, type **. $PGSHOME/bin/*sgiX*/pgs-dev-env.ksh**, press **Return** (note the "dot" and then space at the beginning of this command).

3 This step is optional. Edit the file $HOME/.cshrc and add the line **alias *aliasname* 'setenv PGSHOME *ToolkitPathname*; source $PGSHOME/bin/*sgiX*/pgs-dev-env.csh; echo "*textmessage*" '**.

- The ***aliasname*** is the name of the alias. For example, to set up an environment for the DAAC version of the Toolkit for FORTRAN 77 (or C), you might use **DAACf77** as an ***aliasname***.
- The ***ToolkitPathname*** is the home directory of the particular SDP Toolkit version being used. Refer to Table 26.6.2-1. Note that the setting of PGSHOME shown in this table may differ in your local DAAC.
- The ***sgiX*** is one of: **sgi** for 32-bit version of the Toolkit or **sgi64** for 64-bit version of the Toolkit.
- The ***textmessage*** is a message that will be echoed to the screen signifying that a new Toolkit environment has been set up. It must be enclosed within
- double quotes ("). An example may be, **"DAAC F77 Toolkit environment is now set."**
- A complete example (it should be all on one line in the .cshrc file):

  alias DAACf77 'setenv PGSHOME /$CUSTOM/TOOLKIT/bin/sgi64_daac_f77/; source $PGSHOME/bin/sgi/pgs-dev-env.csh; echo "DAAC F77 Toolkit environment is now set" '

- Other aliases for other versions of the Toolkit can be set up similarly.

26.6.2.4 An example of Compile procedures used to produce a PGE.exe :

1 Setup for PGE07:

```
/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source
rm MOD_PR10.exe
rm *.o
setenv PGSHOME /usr/ecs/OPS/CUSTOM/TOOLKIT/sgi32_daac_f77/
```

```
source $PGSHOME/bin/sgi32/pgs-dev-env.csh
source
/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source/MODIS_setup.csh.pge07
alias
n32_f77
env
make -f MOD_PR10.mk &
ls -l *exe
setenv PGS_PC_INFO_FILE
/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source/MOD_PR10.pcf
ls
MOD_PR10.exe &
confirm execution when done by looking at   file :  vi
MOD_PR10_ClopyL1BmetaToSnow.c
see if job is running:  ps -u emcleod  "time updating for MOD_PR10"
p0spg01{emcleod}88: ps -u emcleod
   PID TTY      TIME CMD
   267 ?       3:13 biod
 25825 pts/11  0:01 csh
 21994 pts/10  0:01 csh
 23215 pts/16  0:00 csh
 26242 pts/10  0:07 MOD_PR10.
 26089 pts/11  0:01 xedit
 26318 pts/10  0:00 ps
p0spg01{emcleod}105: pwd
/tmp_mnt/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source
p0spg01{emcleod}106: ls
MODIS_setup.csh.pge07        MOD_PR10_CopyL1BmetaToSnow.c
MODIS_setup_OPS              MOD_PR10_CopyL1BmetaToSnow.o
MOD_PR10.exe                 MOD_PR10_MakeMeta.c
MOD_PR10.h                   MOD_PR10_MakeMeta.o
MOD_PR10.mcf                 MOD_PR10_Process_Cloud.c
MOD_PR10.mk                  MOD_PR10_Process_Cloud.o
MOD_PR10.pcf                 MOD_PR10_Process_GEO.c
MOD_PR10_AAmain.c            MOD_PR10_Process_GEO.o
MOD_PR10_AAmain.o            MOD_PR10_Process_L1B.c
MOD_PR10_Compute_Snow.c      MOD_PR10_Process_L1B.o
MOD_PR10_Compute_Snow.o      MOD_PR10_Process_SnowFile.c
MOD_PR10_CopyGEOmetaToSnow.c  MOD_PR10_Process_SnowFile.o
MOD_PR10_CopyGEOmetaToSnow.o  compile_smf.csh
p0spg01{emcleod}107:
```

26.6.2.5 Example of a PGE Executables Tar File Insertion Script

This example was produced in Drop 4 and is provided for review only. Go to the section Placing the Science Software Executable (SSEP) on the Data Server which includes the Insertion of a PGE Tar file..

```
Configuration filename?  (enter for default:
../../cfg/EcDpAtInsertExeTarFile.CFG)
ECS Mode of operations?  (enter for default: OPS)
Name of PGE?  (enter for default: PGE07)
Science software version of PGE?  (enter for default: 2)
Staged filename to insert (including FULL path)?  (enter for default:
/home/emcleod/SSEP/PGE07.tar)
Associated ASCII metadata filename to insert (including FULL path)?  (enter for
defaul /home/emcleod/SSEP/PGE07.tar.met)
Top level shell filename within tar file?  (enter for default: PGE07.csh)
PGE07.csh
Warning: Could not open message catalog "oodce.cat"
/usr/ecs//OPS/CUSTOM/bin/DPS/EcDpAtInsertExeTarFile: Process Framework:
ConfigFile ..//cfg/EcDpAtInsertExeTarFile.CFG  ecs_mode OPS
Performing INSERT.....
Retrieved from IOS for ESDT = PGEEXE the DSS UR =
UR:15:DsShSciServerUR:13:[MDC:DSSDSR
Trying to make a request to [MDC:DSSDSRV]
Trying to make a request to [MDC:DSSDSRV]
Insert to Data Server and PDPS database update successful for:
  PGE name = 'PGE07'
  Ssw version = '2'
  ESDT = 'PGEEXE'
  ESDT Version = "001”
  staged file = '/home/emcleod/SSEP/PGE07.tar'
  metadata file = '/home/emcleod/SSEP/PGE07.tar.met'
  Top level shell name = 'PGE07.csh'
Inserted at UR:
'UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:14:LM:PGEEXE:94'
Hit return to run again, 'q <return>' to quit:
```

26.6.3 Compiling Status Message Facility (SMF) Files

Status Message Facility (SMF) files are used by the SDP Toolkit to facilitate a status and error message handling mechanism for use in the science software and to provide a means to send log files, informational messages, and output data files to DAAC personnel or to remote users.

Science software making use of the SMF need particular header (include) files when being built and also need particular runtime message files when being run. Both the header and message files are produced by running a SMF "compiler" on a message text file. These message text files should be part of the science software delivery to the DAAC. They typically have a .t file name extension.

This procedure describes how to compile the SMF message text files to produce both the necessary include files and the necessary runtime message files.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The C shell (or a derivative) is the current command shell.

To check compile status message facility (SMF) files, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SGI.
- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SGI.
- It is recommended that this procedure begin within a new command shell on the SGI.

2 If required, at the UNIX prompt on the SGI, type **cleartool setview *ViewName***, press **Return**.
- The ***ViewName*** is the name of a view allowing the SMF files to be accessible.
- This step is only necessary if any of the SMF files are in ClearCase (in the VOB under configuration management).

3 At the UNIX prompt on the SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.
- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version.
- The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

4 At the UNIX prompt on the SGI, type **cd *pathname***, press **Return**.
- The ***pathname*** is the full path name to the directory containing the SMF text files.
- The SMF text files will typically have .t file name extensions.

5 At the UNIX prompt on the SGI, type **smfcompile -f *textfile*.t -lang** , press **Return**.
- The ***-lang*** is a flag that indicates for what language to compile. This flag can be one of **-c** to produce C header files, **-f77** to produce FORTRAN 77 include files, and **-ada** to produce Ada include files. The default is for C include files. For example, type **smfcompile -f77 PGS_MODIS_39123.t**, press **Return**.
- The ***textfile*** is the file name of the SMF text file delivered with the science software.
- The SMF text files will typically have .t file name extensions.
- File names for SMF text files usually have the "seed" value used by the file as part of its file name (*e.g.* PGS_MODIS_39123.t where 39123 is the seed number).

- Only one such SMF text file can be compiled at a time; wildcards cannot be used.
- The SMF compiler may be run with the additional flags **-r** and **-i** as in, **smfcompile -f *textfile*.t -r -i**. The **-r** automatically places the runtime message file in the directory given by the environment variable PGSMSG. The **-i** automatically places the include file in the directory given by the environment variable PGSINC. For example, type **smfcompile -ada -r -i -f PGS_MODIS_39123.t**, press **Return**. Note that the **-f** flag must always be immediately followed by the name of the text file.

6 If necessary, at the UNIX prompt on the SGI, type **mv *IncludeFilename* $PGSINC**, press **Return**. Then, type **mv *RuntimeFilename* $PGSMSG**, press **Return**.

- This step is only required if either the **-r** or the **-i** flag were not used in step 5.
- The ***IncludeFilename*** is the name of the include file created in step 5.
- The ***RuntimeFilename*** is the name of the runtime message file created in step 5.
- For example, type **mv PGS_MODIS_39123.h $PGSINC**, press **Return**. And then type, **mv PGS_MODIS_39123 $PGSMSG**, press **Return**.

26.6.4 Building Science Software with the SCF Version of the SDP Toolkit

In order to be tested at the DAAC, science software must be compiled and linked to produce binary executables. These binary executables are then packaged into one or more shell scripts as defined by the science software developer (Instrument Team). These science software packages are the Product Generation Executives (PGEs) delivered to the DAACs during SSI&T. PGEs are the smallest schedulable unit of science software in the ECS.

Building science software into PGEs should be done in accordance with supplied documentation. Such documentation should describe the process in detail. In general, science software deliveries will come with make files or other build scripts to automate the build process.

In general, science software will be built, run, and tested with the SCF version of the SDP Toolkit to ensure that the software has been successfully ported to the DAAC. Once this test has been completed successfully, the science software will be re-built, rerun, and re-tested with the DAAC version of the SDP Toolkit. Only with the DAAC Toolkit can the PGE be run within the ECS.

This procedure describes some general principals that may or may not be applicable to a particular science software delivery for building a PGE with the SCF version of the SDP Toolkit. See Section for Building a PGE with the DAAC version of the SDP Toolkit.

Building Science Software with the SCF Version of the SDP Toolkit - Activity Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1 The C shell (or a derivative) is the current command shell.

To build science software with the SCF version of the SDP Toolkit, be aware of the "typical" procedure steps that follow:

1 Read all instructional material supplied with the science software delivery. Such material should be the primary source of information on how to build the science software.

- Read the *Systems Description* document and the *Operations Manual*. Both of these or their equivalent should be in the delivery.
- Typically, there will be "readme" files accompanying each PGE in the directory structure, perhaps in a doc directory.
- Text files (ASCII) may be viewed with the UNIX command, *more* or with the *vi* editor.
- PostScript documents may be viewed with *ghostview*, which is accessible via the SSIT Manager.
- PDF formatted documents may be viewed with *acroread*, the Acrobat Reader, also accessible via the SSIT Manager.
- Documents in Microsoft Word and related formats may be viewed through the Microsoft Windows™ 3.1 emulator. The MS Windows emulator may be accessed from the SSIT Manager.

2 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SGI.
- It is recommended that this procedure begin within a new command shell on the SGI.

3 At the UNIX prompt on the SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, an SCF version.
- The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

4 If make files are in ClearCase, at the UNIX prompt on the SGI, type **cleartool setview *ViewName***, press **Return**. Then, **cd *pathname***, press **Return**. And **cleartool checkout -nc *makefile***, press **Return**.

- The ***ViewName*** is the name of a view allowing the make files to be accessible.
- The ***pathname*** is the full path name of the directory (in the VOB) where the make file has been checked in.
- The ***makefile*** is the name of the make file to examine and possibly modify.
- This step is only necessary if any of the make files (or build scripts) are in ClearCase (in the VOB under configuration management).

5 Examine and alter (if necessary) any make files using any text editor (*vi*, *emacs*).

- There may be several make files for a particular PGE.
- Verify that compiler, compiler flag settings, and other environment variable settings are appropriate.

- The Toolkit set up (from step 3) will set many environment variables which can be used in the make files. To see the current environment variable settings, at the UNIX prompt on the SGI, type **env**, press **Return**.

6 Compile any required status message facility (SMF) files and place the header file(s) in the proper directory for building.

7 Verify that the directory structure for the PGE source files matches the directory structure expected by the make files or build scripts.

- Deliveries may come with install scripts that place files into various directories according to some predefined structure.

8 If necessary, at the UNIX prompt on the SGI, type **cleartool checkout -nc *filename***, press **Return**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked out of ClearCase. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check out step.
- Note that checking in executable or object files is *not* recommended in the first place.

9 Build the software in accordance with instructions delivered.

- Science software deliveries may come with a single, top-level script to do the entire build or the build process could involve a series of steps, each of which should be described fully in the delivered documentation.
- Choose the most appropriate optimization/debugger flag. During testing, the "-g" is often used. This results in larger and slower executables, but assists in debugging. For production, the "-O" flag may be used to optimize execution time. Variants of the "-g" and "-O" flags may be incompatible.

10 If necessary, at the UNIX prompt on the SGI, type **cleartool checkin *filename* -nc**, press **Return**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked into ClearCase. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check in step.
- Note that checking in executable or object files is *not* recommended.

26.6.5 Building Science Software with the DAAC Version of the SDP Toolkit

In general, science software will be built, run, and tested with the SCF version of the SDP Toolkit to ensure that the software has been successfully ported to the DAAC. Once this test has been completed successfully, the science software will be re-built, rerun, and re-tested with the DAAC version of the SDP Toolkit. Only with the DAAC Toolkit can the PGE be run within the ECS.

This procedure describes some general principals that may or may not be applicable to a particular science software delivery for building a PGE with the DAAC version of the SDP Toolkit.

Building Science Software with the DAAC Version of the SDP Toolkit - Activity Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

The C shell (or a derivative) is the current command shell.

To build science software with the DAAC version of the SDP Toolkit, be aware of the "typical" procedure steps that follow:

1 Read all instructional material supplied with the science software delivery. Such material should be the primary source of information on how to build the science software.

- Read the *Systems Description* document and the *Operations Manual*. Both of these or their equivalent should be in the delivery.
- Typically, there will be "readme" files accompanying each PGE in the directory structure, perhaps in a doc directory.
- Text files (ASCII) may be viewed with the UNIX command, *more* or with the *vi* editor.
- PostScript documents may be viewed with *ghostview*, which is accessible via the SSIT Manager.
- PDF formatted documents may be viewed with *acroread*, the Acrobat Reader, also accessible via the SSIT Manager.
- Documents in Microsoft Word and related formats may be viewed through the Microsoft Windows™ 3.1 emulator. The MS Windows emulator may be accessed from the SSIT Manager.

2 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SGI.
- It is recommended that this procedure begin within a new command shell on the SGI.

3 At the UNIX prompt on the SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, a DAAC version.
- The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

4 If make files are in ClearCase, at the UNIX prompt on the SGI, type **cleartool setview *ViewName***, press **Return**. Then, **cd *pathname***, press **Return**. And **cleartool checkout -nc *makefile***, press **Return**.

- The ***ViewName*** is the name of a view allowing the make files to be accessible.
- The ***pathname*** is the full path name of the directory (in the VOB) where the make file has been checked in.
- The ***makefile*** is the name of the make file to examine and possibly modify.

- This step is only necessary if any of the make files (or build scripts) are in ClearCase (in the VOB under configuration management).

5 Examine and alter (if necessary) any make files using any text editor (*vi, emacs*). If the software had already been built and tested with the SCF version of the SDP Toolkit, this step may be unnecessary.

- There may be several make files for a particular PGE.
- Verify that compiler, compiler flag settings, and other environment variable settings are appropriate.
- The Toolkit set up (from step 3) will set many environment variables which can be used in the make files. To see the current environment variable settings, at the UNIX prompt on the SGI, type **env**, press **Return**.

6 Compile any required status message facility (SMF) files and place the header file(s) in the proper directory for building.

7 Verify that the directory structure for the PGE source files matches the directory structure expected by the make files or build scripts.

- Deliveries may come with install scripts that place files into various directories according to some predefined structure.

8 If necessary, at the UNIX prompt on the SGI, type **cleartool checkout -nc *filename***, press **Return**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked out of ClearCase. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check out step.
- Note that checking in executable or object files is *not* recommended in the first place.

9 Build the software in accordance with instructions delivered.

- Science software deliveries may come with a single, top-level script to do the entire build or the build process could involve a series of steps, each of which should be described fully in the delivered documentation.
    1. Choose the most appropriate optimization/debugger flag. During testing, the "-g" is often used. This results in larger and slower executables, but assists in debugging. For production, the "-O" flag may be used to optimize execution time. Variants of the "-g" and "-O" flags may be incompatible.

10 If necessary, at the UNIX prompt on the SGI, type **cleartool checkin *filename* -nc**, press **Return**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked into ClearCase. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check in step.
- Note that checking in executable or object files is *not* recommended.

This page intentionally left blank.

26.7 Running a PGE in a Simulated SCF Environment

Science software delivered to the DAACs for SSI&T was developed and tested at individual SCFs using the SCF version of the SDP Toolkit. Before linking the software with the DAAC version of the Toolkit and integrating it with the ECS, it is prudent to first link the software to the SCF version of the Toolkit and run it as it was run at the SCF. This type of testing can reveal problems associated with the process of porting the software to another platform whose architecture may be quite different from the one on which the software was developed.

A simulated SCF environment means that the software is built using the SCF version of the Toolkit and is run from the UNIX command line. The Planning and Data Processing System (PDPS) and the Data Server are not involved.

The procedures which follow describe how to run the science software in a simulated SCF environment.

26.7.1 Setting Up the Environment for Running the PGE

Running a PGE that has been built with the SCF version of the SDP Toolkit requires some environment set up as it does at the SCF. This procedure describes how to set up a simulated SCF environment.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

- The Process Control File (PCF) exists and has been tailored for the DAAC environment.
- The C shell or a derivative (*e.g.* T shell) is the current user shell.

To set up an environment for running the PGE, execute the procedure steps that follow:

---

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SPR SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SPR SGI.
- It is recommended that this procedure begin within a new command shell on the SPR SGI.

2 At the UNIX prompt on the SPR SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, an SCF version.
- The ***sgiX*** refers to the appropriate processor (see Section 9.2 ?). For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

At the UNIX prompt on the SPR SGI, type **setenv PGS_PC_INFO_FILE *PCFpathname*/*PCFfilename***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the Process Control File (PCF) to be associated with this PGE.

- The ***PCFfilename*** is the file name of the PCF.
- For example, **setenv PGS_PC_INFO_FILE /disk2/PGE32/PCF/PGE32.pcf**, press **Return**.

3 Optionally, at the UNIX prompt on the SPR SGI, type **rm *LogPathname*/*LogFilename***, press **Return**.

- The ***LogPathname*** is the full path name to the location of the PGE log files for this PGE.
- The ***LogFilename*** is the file name of the PGE log file to remove from a previous run of the same PGE. PGE log files can be Status, User, or Report.
- The ***LogFilename*** may use wildcard characters to remove all of the log files at the same time.
- This step is optional. If log files from a previous run of the same PGE are not removed, they will be appended with the information from the current run.
- The environment will then be set up. Continue on the next Section.

4 If necessary, set any other shell environment variables needed by the PGE by sourcing the appropriate scripts or setting them on the command line.

- For example, for a PGE requiring IMSL, at the UNIX prompt on the SPR SGI, type **source /usr/ecs/TS1/COTS/imsl/vni/ipt/bin/iptsetup.cs**, press **Return**
- For some PGEs, the environment variables to be set will be specified in the documentation or the files to source will be supplied in the delivery. Refer to documentation included in the delivery.

26.7.2 Running and Profiling the PGE

Profiling a PGE refers to the process of gathering information about the runtime behavior of a PGE. The information includes the wall clock time, user time and system time devoted to the PGE; the amount of memory used; the number of page faults; and the number of input and output blocks.

The Planning and Data Processing System (PDPS) database must be populated with the above information when the PGE is registered with the PDPS during the integration phase of SSI&T. This information may be delivered with the PGE or it may need to be determined at the DAAC during SSI&T. This procedure addresses the latter need.

Note that profiling, as used here, does not involve altering the binary executable to produce instrumented code.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

- The PGE has been built successfully with the SCF version of the SDP Toolkit .
- The required SMF runtime message files have been produced and placed in the correct locations.

- The Process Control File (PCF) exists and has been tailored for the DAAC environment.
- The required environment for running the PGE has been set up.
- The required input files are available and accessible.
- The C shell or a derivative (*e.g.* T shell) is the current user shell.

To run and profile the PGE, execute the procedure steps that follow:

1 At the UNIX prompt on the SPR SGI in the window containing the set up environment, type **cd *PGEbinPathname***, press **Return**.
- The ***PGEbinPathname*** is the full path name of the directory containing the built PGE binary executable. For example, **cd /disk3/PGE32/bin/**, press **Return**.

2 At the UNIX prompt on the SPR SGI, type **/usr/ecs/<mode>/CUSTOM/bin/DPS/E*cDpPrR*usage *PGE* >& *ResultsOut***, press **Return**.
- The ***PGE*** is the name given to the PGE binary executable.
- The ***ResultsOut*** is the file name in which to capture the profiling results as well as any messages from standard output (stdout) and standard error (stderr) that may be produced by the running PGE. Note that PGEs should *not* write to stdout or stderr.
- The **EcDpPrRusage** is the profiling program that outputs information about the runtime behavior of the PGE.
- Depending upon the PGE, it may take some time before the UNIX prompt returns.

3 At the UNIX prompt on the SPR SGI, type **echo $status**, press **Return**.
- The **$status** is an environment variable that stores the exit status of the previous program run, in this case, the PGE.
- A status of zero indicates success; a status of non-zero indicates an error of some kind.
- The meaning of a non-zero exit status should be documented and included with the DAPs.
- This command must be run ***immediately*** after the **EcDpPrRusage** command.

4 At the UNIX prompt on the SPR SGI, type **vi *ResultsOut***, press **Return**.
- The ***ResultsOut*** is the file name under which the profiling output was saved. Other output of the PGE may also be in this file.
- The **EcDpPrRusage** results may then be recorded and used when the PGE is registered in the PDPS.
- Any text editor/viewer may be used.

Sample of an Rusage File produced:

```
p0spg01{emcleod}6: more Profile.out
    #   source .cshrc
    #   cd TEST/MOD*
    #   ls
    #   /usr/ecs/OPS/CUSTOM/bin/DPS/EcDpPrRusage MOD_PR10.exe > Profile.out
    p0spg01{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
p0spg01{emcleod}10:
```

26.8 File Comparison and Data Visualization

The purpose of File Comparison is to verify that the output files produced at the DAAC are identical (within tolerances) to the test output files delivered with the DAPs. A successful comparison is a strong indication that the porting of the science software from the development facility at the SCF to the operational facility at the DAAC has not introduced any errors.

A number of file comparison tools are available during SSI&T via the SSIT Manager GUI or they can be invoked from the UNIX command line. Two tools are available for comparing HDF or HDF-EOS files, one tool for comparing ASCII files, and another tool for assisting in comparing binary files.

26.8.1 Using the GUI HDF File Comparison GUI

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. Two HDF or HDF-EOS files exist with similar structures.
3. The Instrument Team has delivered test output files.
4. If either of the two HDF/HDF-EOS files is in the ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

Comparing Two HDF or HDF-EOS Files Using the HDF File Comparison GUI

1 From the SSIT Manager, select **Tools→ Product Examination → HDF** from the menu.

The HDF File Comparison GUI window will be displayed.

2 In the HDF File Comparison Tool GUI, click on the **File 1** button.

- Read the Systems Description document and the Operations Manual. Both of these or their equivalent should be in the delivery.

26.8.2 Using the hdiff HDF File Comparison Tool

The hdiff File Comparison Tool is a text-oriented tool run from the command line. It allows comparison of two HDF or HDF-EOS files

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. Two HDF or HDF-EOS files exist with similar structures.
3. The instrument Team has delivered test output files.
4. If either of the two HDF/HDF-EOS files is in the ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

Comparing two HDF or HDF-EOS Files Using the hdiff File Comparison Tool

1 From the SSIT Manager, select **Tools→ Product Examination → File Comparison → HDF** from the menu.

The HDF File Comparison Tool window will be displayed.

- An xterm window running *hdiff* will be displayed.

2 In the xterm window at the prompt **Options? (-h for help)**, type in any desired options then press the **Enter** key.

- To see the list of available options, type **-h** then press the **Enter** key. to the prompt.

3 In xterm window at the prompt **1st file to compare?**, type ***filename1***, then press the **Enter** key.

- The ***filename1*** is the file name of the first of two HDF or HDF-EOS files to be compared.
- If ***filename1*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name.

4 In xterm window at the prompt **2nd file to compare?**, type ***filename2***, then press the **Enter** key.

- The ***filename2*** is the file name of the second of two HDF or HDF-EOS files to be compared. Select another studentsÕ file.
- If ***filename2*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name. The two files will be compared and the output will be displayed in the xterm window.

26.8.3 Using the ASCII File Comparison Tool

Most output files (products) from PGEs run in the DAAC will be in HDF-EOS format. A small minority may be in ASCII (text) format. The ASCII File Comparison Tool is a front-end to *xdiff* UNIX X Window tool for comparing two ASCII files.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. Two ASCII files exist and have read permissions.
3. The instrument Team has delivered test output files.
4. If either of the two ASCII files are in the ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

Comparing Two ASCII Files

1 From the SSIT Manager, select **Tools→ Product Examination → File Comparison → ASCII** from the menu.
An xterm window running *xdiff* will be displayed.

2 In xterm window at the prompt **1st file to compare?**, type ***filename1***, then press the **Enter** key. Select a descriptor or mcf file in the directory with the PGE.
The ***filename1*** is the file name of the first of two ASCII files to be compared.
If ***filename1*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name.

3 In xterm window at the prompt **2nd file to compare?**, type ***filename2***, then press the **Enter** key.
The ***filename2*** is the file name of the second of two ASCII files to be compared. Select another studentÕs corresponding file.

If ***filename2*** is not in the current directory (the directory from which the SSIT Manager was run), include the full path name with the file name.

A window labeled **xdiff** will be displayed.

4 In the window labeled **xdiff**, view the differences between the two files displayed.

File ***filename1*** will be displayed on the left side of the window. File ***filename2*** will be displayed on the right.

Only sections of file in which there are differences will be displayed. A "bang" character (!) at the beginning of a line indicates that a difference was found.

For further help on *xdiff*, type **man xdiff**, in an xterm window then press the **Enter** key.

Close the display window by using the pull down menu from the X window in the upper left corner.

5 In the xterm window at the prompt **Hit Enter for another diff, 'q <Enter>' to quit:**, type **q** press **Enter** to quit or just press **Enter** to perform another comparison.

26.8.4 Using the Binary File Difference Assistant

Most output files (products) from PGEs run in the DAAC will be in HDF-EOS format. A small minority may be in some binary format. The Binary File Difference Assistant aids the user in constructing code that allows comparison of binary output files. Since there is an unwieldy number of possibilities for binary file formats, this tool cannot compare two binary files without some custom code written at the DAAC, hence, the "Assistant" in the name. The Binary File Difference Assistant aids the user by generating a makefile, a driver module, and a template comparison module in C, FORTRAN 77 or IDL (Interactive Data Language). The user then edits these templates to read the particular binary format in question according to a SCF-supplied format specification.

The binary file comparison will not be performed during the SSIT training lesson.

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. Two Binary files exist and have read permissions.
3. The instrument Team has delivered test output files.
4. If either of the two Binary files are in the ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

Comparing two Binary Files

---

1 From the SSIT Manager, select **Tools→ Product Examination → File Comparison → Binary** from the menu.

The Binary File Difference Assistant tool GUI will be displayed.

2 In the Binary File Difference Assistant tool GUI, click on one of the languages listed under the **Select Language** label. The choices are C, FORTRAN, or IDL.

The choice of language depends largely on preference. It does not necessarily have to be the language that was used to create the files being compared.

3 Optionally, click on either the **Image** button or the **Structure** button located under the label **Compare Function**.

Clicking on the **Image** button will display a code example for comparing binary files containing images.

Clicking on the **Structure** button will display a code example for comparing binary files containing structures or records.

The displayed listing well documented and should be read.

The language of the code will depend on the language selection made in step 2.

4 Optionally, click on either the **Image** button or the **Structure** button located under the label **Driver**.

Clicking on the **Image** button will display a code example for a driver invoking the compare function for binary files containing images.

Clicking on the **Structure** button will display a code example for a driver invoking the compare function for binary files containing structures or records.

The displayed listing well documented and should be read.

The language of the code will depend on the language selection made in step 2.

5 Optionally, click on either the **Help** button.

A Help window will be displayed.

To end help, click on the **Dismiss** button.

The Help window may remain displayed while using the Binary File Difference Assistant.

6 Once familiar with the code examples (steps 3 and 4), click on the **Copy** button.

- A window labeled **Enter Unique ID** will be displayed.
- In the field labeled **Enter unique file identifier:**, type ***fileID***, click on the **OK** button.
- The ***fileID*** will be used in the file names of the files copied over. These files will be:

C:

DaacBinDiff_***fileID***.c	Compare function
DaacBinDiff_***fileID***_driver.c	Driver
DaacBinDiff_***fileID***.mak	Makefile

FORTRAN:

DaacBinDiff_***fileID***.f	Compare function
DaacBinDiff_***fileID***_driver.f	Driver
DaacBinDiff_***fileID***.mak	Makefile

IDL:

DaacBinDiff_***fileID***.pro	Compare function
DaacBinDiff_***fileID***_driver.pro	Driver

DaacBinDiff_***fileID***.sh — Shell script with here document

- The files will be copied into the directory from which the SSIT Manager is being run.

1. Using any desired text editor, customize the files for the job at hand. Then build the executable using the customized makefile provided (for C and FORTRAN). Then run the program to perform the binary file comparison.

26.8.5 Data Visualization

In order to view the success of science software in producing scientifically valid data sets, the data needs to be displayed in forms that convey the most information. Data visualization enables this to be done.

There are two visualization tools provided to the DAAC: EOSView and Interactive Data Language (IDL). These tools are both accessible via the SSIT Manager. EOSView is user friendly GUI for creating two-dimensional displays from HDF-EOS objects(Grid, Swath) as well as the standard HDF objects (SDS, Vdata, Image, Text). It has additional features such as thumbnail-panning, colorization, zooming, plotting, and animation. Only some aspects of data visualization will be addressed in this training material. For further information, see the related references.

IDL is a COTS display and analysis tool widely applied in the scientific community, It is used to create two-dimensional, three dimensional (volumetric), and surface/terrain displays from binary, ASCII, and many other formats in addition to HDF.

Only a limited number of file types will be available during SSIT training.

26.8.5.1 Data Visualization with EOSView

26.8.5.1.1 Viewing Product Metadata with the EOSView Tool

This procedure describes how to use the EOSView tool to inspect the metadata in the HDF-EOS output file from a PGE. To view product metadata with the EOSView tool, execute the procedure steps that follow:

Viewing Product Metadata ** The SSIT Manager does not support Data Visulization at the time of Drop 4.***** A backup example is as follows for EOSView:**

Log into an Algorithm and Test Tools (AITTL) environment using using a machine so configured. At the mini-daac this machine is **p0ais01**

1 Telnet into **p0ais01.**

2 **logon using your own ID and Password**

3 **cd /usr/ecs/TS1/CUSTOM/eosview.**

4 Select EOSView, The EOSView GUI will be displayed.

5 Use the select buttons to guide you toward the view desired

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.

Viewing Product Metadata

1 From the SSIT Manager, select **Tools→ Product Examination → EOSView** from the menu.

- The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Open.** The **Filter** GUI will be displayed.

- In the subwindow labeled **Filter**, select the appropriate directory and file to open.

3 A GUI labeled **EOSView - *MyOutputFile.hdf*** will be displayed where *MyOutputFile.hdf* is the file name of the file chosen in step 2. Once displayed, a list of HDF objects will appear in the main window. If nothing is listed, it means that no HDF objects were found within the file.

4 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on an object listed for which metadata is to be inspected. The object selected will be highlighted.

- Do not double click on object since this will cause a **Dimension** GUI to be displayed instead.

5 The global metadata associated with the object selected will be displayed in a scrollable field by clicking on the **Attributes** menu and selecting **Global** in the GUI labeled **EOSView - *MyOutputFile.hdf***.

- If instead, the message "Contains no Global Attributes" appears, then the selected object contains no global metadata.

6 Repeat steps 4 and 5 for each HDF object within the selected HDF-EOS file for which metadata is to be examined.

7 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **File** menu and select **Close** to close the **EOSView - *MyOutputFile.hdf*** GUI.

8 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Exit** to exit **EOSView - EOSView Main Window** GUI.

26.8.5.1.2 Viewing HDF Image Objects

This procedure describes how to use the EOSView tool to view science Images in the HDF-EOS output file from a PGE.

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.
3. At least one object is an HDF image (RIS8, RIS24, *i.e.* Browse data).

Viewing HDF Image Objects

1 From the SSIT Manager, select **Tools→ Product Examination → EOSView** from the menu.
The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on an Image object listed for which data is to be inspected.
A GUI labeled **EOSView - Image Display Window - *MyImageObject*** will be displayed where *MyImageObject* is the name of the object selected.

3 Optional colorization. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Palette** menu, then select one of the palettes listed: **Default**, **Greyscale**, **Antarctica**, **Rainbow**, or **World Colors**.
This selection may be repeated until the desired palette is chosen.

4 Optional zooming. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Zooming** menu, then select and then select one of the resampling methods listed: **Bilinear Interpolation** or **Nearest Neighbor**. Then click on the **Zoom In** or **Zoom Out** buttons to apply the method.

5 Optional panning while zooming. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Options** menu, then select **Pan Window**, a thumbnail representation of the entire Image will be displayed in the subwindow labeled **Pan Window**. The portion of the zoomed Image shown in the main window will be the portion indicated by the hollow rectangle on the thumbnail image. Use the mouse left button to click and drag the rectangle to a new location on the thumbnail image.
The panning option may be repeated as desired.

6 To end the session with colorization, zooming, or panning, in the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **File** menu and select **Close**.

7 Optional animation. In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **Options** menu, then select **Animated images**.
A GUI labeled EOSView - Image Animation Window - *MyOutputFile.hdf* will be displayed.
Optionally, click on the **Options** menu select **Mode** to select how the animation is to be run. Choose **Stop at end, Continuous run, or Bounce**.

To end animation session, click on the **File → Close.**

26.8.5.1.3 Viewing HDF-EOS Grid Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF-EOS Grid format. These are generally the science data and not browse images.

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.
3. At least one object is an HDF-EOS Grid.

Viewing HDF-EOS Grid Objects

1 From the SSIT Manager, select **Tools→ Product Examination → EOSView** from the menu.
The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on an Grid object listed for which data is to be inspected. A GUI labeled **EOSView - Grid Select** will be displayed.
Information on **Grid Information, Projection Information, Dimensions, Attributes** for the selected object can be displayed by clicking on the appropriate checkboxes.

3 In the GUI labeled **EOSView - Grid Select**, click on the **Data Fields** checkbox and then click on the **OK** button. Then double click on one of the data fields listed.
- A GUI labeled **EOSView - Grid - *GridObjectName* - Start/Stride/Edge** will be displayed where *GridObjectName* will be replaced by the name of the Grid object selected in step 1.

4 To display the data in the form of a table of values, in the GUI labeled **EOSView - Grid - *GridObjectName* - Start/Stride/Edge**, click on the checkboxes for both **YDim** and **XDim** and then click on the **OK** button.
- A GUI labeled ***MyDataField*** will be displayed where *MyDataField* will be replaced by the name of the data field selected in step 2.

5 To display the data field in image form, in the GUI labeled ***MyDataField***, click on the **File** menu and then select **Make Image**. A GUI labeled **EOSView - Swath/Grid Image** will appear,

6 Optional colorization, zooming, panning while zooming can be used to obtain your desired output.

3 To end the session with displaying Grid object, in the GUI labeled **EOSView - Swath/Grid**, click on the **File** menu and select **Close**. The **EOSView - Swath/Grid** GUI will disappear.

26.8.5.1.4 Viewing HDF-EOS Swath Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF-EOS Swath format. These are generally the science data and not browse images.

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.
3. At least one object is an HDF-EOS Swath.

Viewing HDF-EOS Swath Objects

1 From the SSIT Manager, select **Tools→ Product Examination → EOSView** from the menu.
The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a Swath object listed for which data is to be inspected.
A GUI labeled **EOSView - Swath Select** will be displayed.
Information on **Dimensions, Geolocation Mappings, Indexed Mappings, Geolocation Fields**, **Attributes** for the selected Swath Object can be displayed by clicking on the corresponding checkboxes.

3 In the GUI labeled **EOSView - Swath Select**, click on the **Data Fields** checkbox and then click on the **OK** button. Then double click on one of the data fields listed.
A GUI labeled **EOSView - Swath - *SwathObjectName* - Start/Stride/Edge** will be displayed where *SwathObjectName* will be replaced by the name of the Swath object selected in step 1.

4 To display the data in the form of a table of values, in the GUI labeled **EOSView - Swath - *SwathObjectName* - Start/Stride/Edge**, click on the checkboxes for both **ScanLineTra** and **PixelsXtrac** and then click on the **OK** button.

5 To display the data field in image form, in the GUI labeled ***MyDataField***, click on the **File** menu and then select **Make Image**.
A GUI labeled **EOSView - Swath/Grid Image** will appear.

6 Optional colorization, zooming, panning while zooming features can be used in the GUI labeled **EOSView - Swath/Grid Image** to obtain your desired image.

– To end the session with displaying Swath object, in the GUI labeled **EOSView - Swath/Grid**, click on the **File → Close**.

26.8.5.1.5 Viewing HDF SDS Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF SDS (standard HDF science data set) format. To view an HDF SDS object with the EOSView tool, execute the procedure steps that follow:

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.
3. At least one object is an HDF-SDS.

Viewing HDF SDS Objects

1 From the SSIT Manager, select **Tools→ Product Examination → EOSView** from the menu.

The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a SDS object listed for which data is to be inspected.

A GUI labeled **EOSView - Multi-Dimension SDS** will be displayed.

A number of checkboxes will be displayed, one for each of the dimensions in the selected SDS (there will be at least two, an X and a Y).

3 In the GUI labeled **EOSView - Multi-Dimension SDS**, click on two of the dimension checkboxes and then click on the **Table** button. Then double click on one of the data fields listed.

A GUI labeled ***MySDS*** will be displayed where *MySDS* will be replaced by the name of the SDS object selected in step 1.

4 To display the data field in image form, in the GUI labeled ***MySDS***, click on the **File** menu and then select **Make Image**.

A GUI labeled EOSView - Image Display Window - *MySDS* will appear,

5 Optional colorization, zooming, panning while zooming can be used to obtain your desired output.

6 To end the session with displaying Swath object, in the GUI labeled **EOSView - Image Display Window - *MySDS***, select **File → Close** from the menu.

- The **EOSView - Image Display Window - *MySDS*** GUI will disappear.

26.8.5.2 Data Visualization with the IDL Tool

26.8.5.2.1 Viewing Product Data with the IDL Tool

The following procedures describe how to use the IDL (Interactive Data Language) COTS tool to inspect the data in the output file from a PGE. These procedures are geared toward binary and ASCII formats, but can be extended to other formats supported by IDL including HDF, NetCDF, and PGE. Consult the IDL references for details on these other formats.

The major activities addresses here include creating an image display, saving an image display, creating a plot display, and saving a plot display.

The following is a list of tools, and or assumptions:
1.The SSIT Manager is running.
2.The output file is binary, ASCII, or one of the other IDL supported data formats.
3.IDL has been properly installed and is accessible to the user.

Viewing Product Data with the IDL Tool

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 Select the procedure depending upon the activity to perform.

3 To end the IDL session, close any display windows remaining, then at the IDL prompt type **quit**, then press the **Enter** key.
The IDL session will be closed.

26.8.5.2.2 Creating an Image Display Using IDL

The following procedure describes how to use the IDL Tool to create an image display.

The following is a list of tools, and or assumptions:
1.The SSIT Manager is running.
2.The PGE output file to be examined is of an IDL-supported type/format (if in doubt, consult the IDL Reference Guide
3.IDL has been properly installed and is accessible to the user.
4.For binary files, data is assumed to be 8-bit characters

Creating an Image Display Using the IDL Tool - Binary Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENR,1,'MyBinaryFilename'***, press the **Enter** key.
The ***MyBinaryFilename*** is the full path name and file name of the binary data file of known dimensions to read in.

The single quotes (‘) must be included around the path/file name.
The **1** is the logical unit number.

3 At the IDL prompt, type ***MyImage=BYTARR(dim1, dim2)***, press the **Enter** key.
The ***MyImage*** is the name to be given to the image once created.
The ***dim1*** and ***dim2*** are the dimensions of the input data.

4 At the IDL prompt, type ***READU,1,MyImage***, press the **Enter** key.

5 At the IDL prompt, type ***TV,MyImage***, press the **Enter** key.
The image, ***MyImage***, should then be displayed.

6 At the IDL prompt, type ***LOADCT,3***, press the **Enter** key.
This command loads color table number 3. Other color tables are available

7 At the IDL prompt, type ***CLOSE,1***, press the **Enter** key.
This closes logical unit 1.
Always close logical units or an error will result the next time an access is attempted.

Creating an Image Display Using the IDL Tool - ASCII Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENR,1,(‘MyASCIIfilename’)***, then press the **Enter** key.

The **MyASCIIfilename** is the full path name and file name of the ASCII data file of known dimensions to read in.
The single quotes (‘) must be included around the path/file name.
The **1** is the logical unit number.

3 At the IDL prompt, type ***MyImage=BYTARR(dim1,dim2)***, then press the **Enter** key.

The ***MyImage*** is the name to be given to the image once created.
The ***dim1*** and ***dim2*** are the dimensions of the input data.

4 At the IDL prompt, type ***READF,1,MyImage***, then press the **Enter** key.

5 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.

The image, **MyImage**, is displayed.

6 At the IDL prompt, type ***LOADCT,3***, then press the **Enter** key.

This command loads color table number 3. Other color tables are available; refer to the IDL Reference Guide for more details.

7 At the IDL prompt, type ***CLOSE,1***, then press the **Enter** key.

This closes logical unit 1.
Always close logical units or an error will result the next time an access is attempted.

Creating an Image Display Using the IDL Tool - PGM Data:

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***READ_PPM,"MyPGMfilename",MyImage,r,g,b***, then press the **Enter** key.
The MyPGMfilename is the full path name and file name of the PGM formatted data file.
The double quotes (“) must be included around the path/file name.
The MyImage is the name to be given to the image created.

3 At the IDL prompt, type ***TVLCT,r,g,b***, then press the **Enter** key.
Note that r,g,b color table syntax is used for most formatted file types in IDL.

4 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.
The image, **MyImage**, should then be displayed.

26.8.5.2.3 Saving an Image Display Using IDL

The next procedure describes how to save an image display (once created) to either a data file or a graphic file.

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running, IDL is running
2. The PGE output file to be examined is of an IDL-supported type/format (if in doubt, consult the IDL Reference Guide
3. For binary files, data is assumed to be 8-bit characters
4. The image display is to be saved in a binary (8-bit) or ASCII (comma-delimited characters) format.

Save an image display using IDL - Binary Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENW,1,('MyBinaryFilename.bin'***), then press the **Enter** key.

The ***MyBinaryFilename.bin*** is the full path name and file name of the binary data file to write out.

The single quotes (') must be included around the path/file name.

The **1** is the logical unit number.

3 At the IDL prompt, type ***WRITEU,1,MyImage***, then press the **Enter** key.

The ***MyImage*** is the name of the image to save.

4 At the IDL prompt, type ***CLOSE,1***, then press the **Enter** key.

This closes logical unit 1.

Always close logical units or an error will result the next time an access is attempted.

26.8.5.2.4 Save an image display using IDL - ASCII Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENW,1,('MyASCIIfilename.asc')***, then press the **Enter** key.

The ***MyASCIIfilename.asc*** is the full path name and file name of the binary data file to write out.

The single quotes (') must be included around the path/file name.

The **1** is the logical unit number.

3 At the IDL prompt, type ***PRINTF,1,MyImage***, then press the **Enter** key.

The **MyImage** is the name of the image to save.

4 At the IDL prompt, type **CLOSE,1**, then press the **Enter** key.

This closes logical unit 1.

Always close logical units or an error will result the next time an access is attempted.

26.8.5.2.5 Save an image display using JPEG Data

1 At the IDL prompt, type **WRITE_JPEG,*"MyJPEGfilename.jpg",MyImage***, press **Return**.

- The ***MyJPEGfilename.jpg*** is the full path name and file name of the JPEG data file to write out.

26.8.5.2.6 Creating a Plot Display Using IDL

The procedures for creating a plot display are clearly described in the IDL manuals; some exceptions are clarified below.

Setting axis limits for a plot:

1 At the IDL prompt, type **SURFACE,MyPlot,AX=70,AZ=70,xrange=[0,20],yrange=[0,20]zrange=[0,30]**, and press **Return**.

- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).
- ***AX*** sets the displayed rotation about the X axis.
- ***AZ*** sets the displayed rotation about the Z axis.
- The values of ***xrange*** set the displayed portion of the X axis.
- The values of ***yrange*** set the displayed portion of the Y axis.
- The values of ***zrange*** set the displayed portion of the Z axis.
- The plot will then be displayed to the screen.

26.8.5.2.7 Setting axis titles for a plot:

1 At the IDL prompt, type **SURFACE,MyPlot,AX=70,AZ=70,xtitle='this is X', ytitle='this is Y',ztitle='this is Z'**, and press **Return**.

- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).
- The value of ***xtitle*** sets the displayed title of the X axis.
- The value of ***ytitle*** sets the displayed title of the Y axis.
- The value of ***ztitle*** sets the displayed title of the Z axis.
- The plot will then be displayed to the screen.

26.8.5.2.8 Saving a Plot Display Using IDL

Saving a displayed plot to a permanent file:

1 At the IDL prompt, type **MyPlotDisplay=SURFACE,MyPlot,AX=80,AZ=20**, and press **Return**.

- The ***MyPlotDisplay*** is session name for the displayed plot of ***MyPlot***.
- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).

2 At the IDL prompt, type **SAVE,MyPlotDisplay,4,'MyPlotOutput.ps'**, press **Return**.

- The ***MyPlotDisplay*** is the session name of the plot display .
- The ***MyPlotOutput.ps*** is the desired name for the saved file.

3. The SAVE option number 4 sets the output file type to PostScript (ps). There are other options, of course (consult the IDL manuals).

26.8.5.3 Raster Mapping Fundamentals

This procedure describes how to use the IDL Tool to perform basic raster mapping functions. These are spatial functions involving map projections, but do not include surface modeling (also called "2.5D") or two-dimensional spectral functions.

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. IDL is running

26.8.5.3.1 Raster Mapping - Global Data Set Image

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.
The ***MyImage*** is the image name of the global image data set.
The image, ***MyImage***, should then be displayed.

3 At the IDL prompt, type ***MAP_SET,/ORTHOGRAPHIC***, then press the **Enter** key.
IDL also supports other map projections. Refer to IDL Reference Guide.

4 At the IDL prompt, type ***MyNewImage=MAP_IMAGE(MyImage,startx,starty,/BILIN)***, then press the **Enter** key.
The ***MyNewImage*** is the name to assign to the resulting image.
The ***MyImage*** is the name of the original global image data set.

5 At the IDL prompt, type ***TV,MyNewImage,startx,starty***, then press the **Enter** key.
The image ***MyNewImage*** should then be displayed.

6 Optional overlay Lat/Long. At the IDL prompt, type ***MAP_GRID***, then press the **Enter** key.
This overlays Lat/Long graticule onto ***MyNewImage***.

7 Optional overlay world coastlines. At the IDL prompt, type ***MAP_CONTINENTS***, press the **Enter** key.
This overlays world coastlines onto **MyNewImage**.

For a sub-global data set image, one having geocentric-LLR coordinates defined for subintervals of longitude and latitude (e.g. from -88 to -77 degrees East Longitude and 23 to 32 degrees North Latitude.

The following is a list of tools, and or assumptions:
1.The SSIT Manager is running.
2.IDL is running

26.8.5.4 Raster Mapping - Sub-Global Data Set Image

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.

The ***MyImage*** is the image name of the sub-global image data set.

The image, ***MyImage***, should then be displayed.

3 At the IDL prompt, type ***MAP_SET,/MERCATOR,LIMIT=[lat1,lon1,lat2,lon2]***, then press the **Enter** key.

The ***lat1***, ***lon1***, ***lat2***, and ***lon2*** specify the latitude and longitude intervals of the sub-global image data set.

4 At the IDL prompt, type ***MyNewImage=MAP_IMAGE(MyImage,startx,starty,/BILIN.LATMIN=lat1,LATMAX=lat2,LONMIN=lon1,LONMAX=lon2)***, then press the **Enter** key.

The ***MyNewImage*** is the name to assign to the resulting image.

The ***MyImage*** is the name of the original global image data set.

The ***lat1***, ***lon1***, ***lat2***, and ***lon2*** specify the latitude and longitude intervals of the sub-global image data set.

5 At the IDL prompt, type ***TV,MyNewImage,startx,starty***, then press the **Enter** key.

The image ***MyNewImage*** should then be displayed.

6 Optional overlay Lat/Long. At the IDL prompt, type ***MAP_GRID***, then press the **Enter** key.

This overlays Lat/Long graticule onto ***MyNewImage***.

7 Optional overlay world coastlines. At the IDL prompt, type ***MAP_CONTINENTS***, then press the **Enter** key.

This overlays world coastlines onto ***MyNewImage***.

26.9 Science Software Integration and Test (SSI&T) Release RELEASE 4.

The process of SSI&T or integration of EOS Instrument Science Software into the ECS has been developed and refined over three iterations of ECS. IR1, the Pre-Release B Testbed and Release RELEASE 4. The latter will be the at-launch system supporting EOS-AM1 instruments. Although every attempt has been made to keep integration procedures for science software consistent through succeeding releases, basis architectural differences have led to significant variances. This section describes the architecture of the last iterations of ECS.

26.9.1 RELEASE 4 Architecture: Overview

The Release RELEASE 4 architecture can be grouped into the following four categories:

- Data storage and management is provided by the Data Server Subsystem (DSS), with the functions needed to archive science data, search for and retrieve archived data, manage the archives, and stage data resources needed as input to science software or resulting as output from their execution. The Data Server Subsystem provides access to earth science data in an integrated fashion through an Application Programming Interface that is common to all layers.
- Information search and data retrieval is provided by the science user interface functions in the Client Subsystem (CLS), by information search support functions in the Data Management Subsystem (DMS), and by capabilities in the Interoperability Subsystem (IOS) which assist users in locating services and data.
- Data processing is provided by the Data Processing Subsystem (DPS) for the science software; and by capabilities for long and short term planning of science data processing, as well as by management of the production environment provided by the Planning Subsystem (PLS). Routine data processing and re-processing will occur in accordance with the established production plans. In addition ECS will provide “on-demand processing”, where higher level products are produced only when there is explicit demand for their creation.
- Data ingest is provided by the Ingest Subsystem (INS), which interfaces with external applications and provides data staging capabilities and storage for an approximately 1-year buffer of Level 0 data (so that reprocessing can be serviced from local storage).The number of external interfaces which ECS will have is potentially very large, and the interfaces can serve very diverse functions, such as high-volume ingest of level 0 data and low-volume ingest of data from field campaigns.

Table 26.9.3-1 provides procedural differences: Testbed to RELEASE 4.

26.9.1.1 ECS Subsystems

The following sub-sections provide brief overviews for each of these subsystems. More detailed discussions of their design breakdown can be found in 305-CD-020-002.

26.9.1.2 Client Subsystem (CLS)

The Client provides users with an interface through which they can access ECS services and data. It also gives science software access to the ECS services, as well as direct access to ECS data. Access is provided through graphic user interface (GUI) application tools for displaying the various kinds of ECS data (e.g., images, documents, tables), and libraries representing the client APIs to ECS services. The client subsystem follows an object oriented design. The design is built around a core set of ‘root’ objects from which all other software will inherit its behavior.

26.9.1.3 Interoperability Subsystem (IOS)

The Interoperability subsystem provides an advertising service. It maintains a database of information about the services and data offered by ECS, and provides interfaces for searching this database and for browsing through related information items. For example, ESDTs are made visible through the advertising service. The Client Subsystem provides the user interface which enables access to the IOS.

26.9.1.4 Data Management Subsystem (DMS)

The Data Management subsystem provides three main functions:

- Provide end-users with a consolidated logical view of a distributed set of data repositories.
- Allow end-users to obtain descriptions for the data offered by these repositories. This also includes descriptions of attributes about the data and the valid values for those attributes.
- Provide data search and access gateways between ECS and external information systems.

26.9.1.5 Data Server Subsystem (DSS)

The Data Server subsystem provides the management, cataloging, access, physical storage, distribution functions for the ECS earth science data repositories, consisting of science data and their documentation. The Data Server provides interfaces for other ECS subsystems which require access to data server services. The Data Server Subsystem consists of the following principal design components:

- Database Management System - The Data Server subsystem will use database technology to manage its catalog of earth science data, and for the persistence of its system administrative and operational data.
- Document Management System - Web server and database technology are used to implement a document management system to provide storage and information retrieval for guide documents, science software documentation, and ECS earth science related documents.
- Data Type Libraries - The Data Server will use custom dynamic linked libraries (DLLs) to provide an extensible means of implementing the variety of ECS earth science data types and services, and will provide a consistent interface for use by other ECS subsystems requiring access to those services and data.
- File Storage Management System - This component provides archival and staging storage for data.
- Distribution System - The Data Server provides the capabilities needed to distribute bulk data via electronic file transfer or physical media.

26.9.1.6 Ingest Subsystem (INS)

This subsystem deals with the initial reception of all data received at an EOSDIS facility and triggers subsequent archiving and processing of the data. The ingest subsystem is organized into a collection of software components (e.g., ingest management software, translation tools, media handling software) from which those required in a specific situation can be readily configured. The resultant configuration is called an ingest client. Ingest clients can operate on a continuous basis to serve a routine external interface; or they may exist only for the duration of a specific ad-hoc ingest task. The ingest subsystem also standardizes on a number of possible application protocols for negotiating an ingest operation, either in response to an external notification, or by polling known data locations for requests and data.

26.9.1.7 Data Processing Subsystem (DPS)

The main components of the data processing subsystem - the science algorithms or Product Generation Executives (PGEs) - will be provided by the science teams. The data processing subsystem provides the necessary hardware resources, as well as a software environment for queuing, dispatching and managing the execution of these algorithms. The processing environment will be highly distributed and will consist of heterogeneous computing platforms. The AutoSys COTS tool is used as the scheduling engine. The tool is designed to manage production in a distributed UNIX environment. The DPS also interacts with the DSS to cause the staging and de-staging of data resources in synchronization with processing requirements.

26.9.1.8 Planning Subsystem (PLS)

The Planning Subsystem provides the functions needed to plan routine data processing, schedule on-demand processing, and dispatch and manage processing requests. The subsystem provides access to the data production schedules at each site, and provides management functions for handling deviations from the schedule to operations and science users. The Planning subsystem provides several functions to account for:

- a processing environment which eventually will be highly distributed and consist of heterogeneous computing platforms
- existence of inter-site and external data dependencies
- dynamic nature of the data and processing requirements of science algorithms
- need for high availability
- providing a resource scheduling function which can accommodate hardware technology upgrades
- support for on-demand processing (as an alternative to predominantly routine processing)
- ability to provide longer-term (e.g., monthly) processing predictions as well as short term

(e.g., daily) planning and scheduling

26.9.1.9 Communications Subsystem (CSS)

The CSS helps manage the operation of distributed objects in ECS, by providing a communications environment. The environment allows software objects to communicate with each other reliably, synchronously as well as asynchronously, via interfaces that make the location of a software object and the specifics of the communications mechanisms transparent to the application.

In addition, CSS provides the infra-structural services for the distributed object environment. They are based on the Distributed Computing Environment (DCE) from the Open Software Foundation (OSF). DCE includes a number of basic services needed to develop distributed applications, such as remote procedure calls (rpc), distributed file services (DFS), directory and naming services, security services, and time services.

Finally, CSS provides a set of common facilities, which include legacy communications services required within the ECS infrastructure and at the external interfaces for file transfer, electronic mail, bulletin board and remote terminal support. The Object Services support all ECS applications with inter-process communication and specialized infra- structural services such as security, directory, time, asynchronous message passing, event logging, lifecycle service, transaction processing and World Wide Web (WWW) service.

26.9.1.10 Management Subsystem (MSS)

The Management Subsystem (MSS) provides enterprise management (network and system management) for all ECS resources: commercial hardware (including computers, peripherals, and network routing devices), commercial software, and custom applications. With few exceptions, the management services will be fully decentralized, such that no single point of failure exists.

MSS provides two levels of an ECS management view: the local (site/DAAC specific) view, provided by Local System Management (LSM), and the enterprise view, provided by the Enterprise Monitoring and Coordination (EMC) at the SMC. Enterprise management relies on the collection of information about the managed resources, and the ability to send notifications to those resources. For network devices, computing platforms, and some commercial of the shelf software, MSS relies on software called “agents” which are usually located on the same device/platform and interact with the device’s or platform’s control and application software, or the commercial software product. However, a large portion of the ECS applications software is custom developed, and some of this software - the science software - is externally supplied. For these components, MSS provides a set of interfaces via which these components can provide information to MSS (e.g., about events which are of interest to system management such as the receipt of a user request or the detection of a software failure). These interfaces also allow applications to accept commands from MSS, provided to MSS from M&O consoles (e.g., an instruction to shut down a particular component). Applications which do not interact with MSS directly will be monitored by software which acts as their “proxies”. For example, the Data

Processing Subsystem (DPS) acts as the proxy for the science software it executes. DPS notifies MSS of events such as the dispatching or completion of a PGE, or its abnormal termination.

MSS uses HP OpenView as the centerpiece of its system management solution. The information collected via the MSS interfaces from the various ECS resources is consolidated into an event history database, some on a near real-time basis, some on a regular polling basis (every 15-to 30 minutes) as well as on demand, when necessitated by an operator inquiry. The database is managed by Sybase, and Sybase query and report writing capabilities will be used to extract regular and ad-hoc reports from it. Extracts and summaries of this information will be further consolidated on a system wide basis by forwarding it to the SMC (also on a regular basis).

MSS provides fault and performance management and other general system management functions such as security management (providing administration of identifications, passwords, and profiles); configuration management for ECS software, hardware, and documents; Billing and Accounting; report generation; trending; request tracking; and mode management (operational, test, simulation, etc.).

26.9.1.11 Internetworking Subsystem (ISS)

The ISS provides local area networking (LAN) services at ECS installations to interconnect and transport data among ECS resources. The ISS includes all components associated with LAN services including routing, switching, and cabling as well as network interface units and communications protocols within ECS resources.

The ISS also provides access services to link the ECS LAN services to Government-furnished wide-area networks (WANs), point-to-point links and institutional network services. Examples include the NASA Science Internet (NSI), Program Support Communications Network (PSCN), and various campus networks “adjoining” ECS installations.

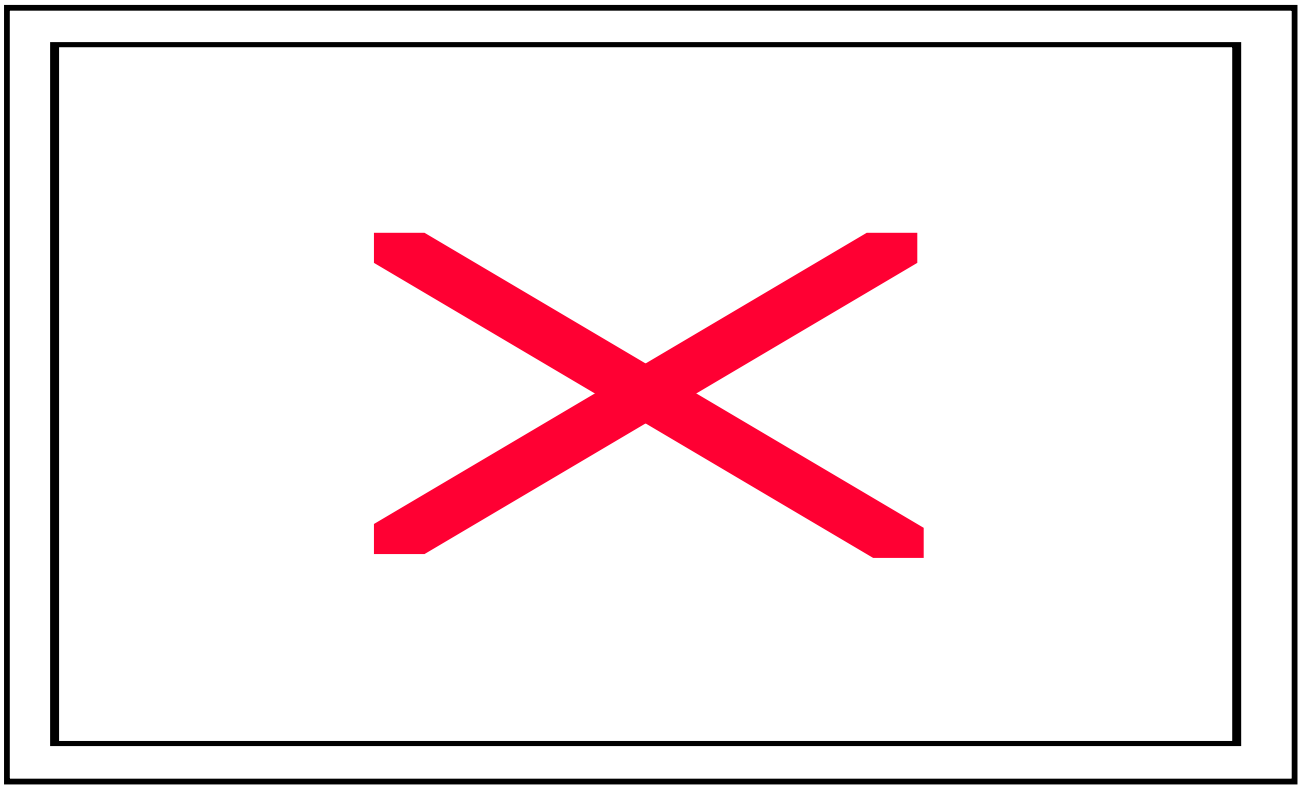


Figure 26.9.1-1. ECS Communications /Interetworking Subsystems

26.9.2 Implications for SSI&T Functions.

Table 26.9.3-1 . List the major Functions that will be encountered by SSI&T staff. These architectural functions characterize the RELEASE 4 systems. The major functions are due to the presence of Ingest and Data Server Subsystems in RELEASE 4.

Function		**Release RELEASE 4**
System Operation		All servers must run and communicate with each other; bring up manually, or use ECS Assist tool.
Ingest Ancillary Data Granules		Ingest GUI, ESDTs must be visible to ADV server.
ESDT Insert		Use Ingest
ESDT Verification		verify through ADV

DAP, SSAP Insert		Use Ingest
PDPS Database Population		More attributes, production rules
PGE Operation		When all data is available; DPR activated. No automatic reprocessing Complex chaining through production rules.
File Access		verify presence through ADV; ftp from SDSRV; access to multiple sites
Multi-file Granule Support		Files inserted together, accessed as a single granule.
Subscription Management		Subscription Manager

Table 26.9.3.1. Major SSI&T Functions within VERSION 2.0

26.10 Using ECS Assistant to View ECS Science Data Server Database

ESDTs and their granules stored in the archive are managed using an ECS Science Data server database. ECS Assistant provides an easy way to review the records stored in this database by using the ECS Assistant DB Viewer. There are two main windows in the DB Viewer. The first is called Collections and is used to display ESDT information included in the Collection database table. Information listed in this table includes ESDT short names, times last updated, types, etc. If an ESDT is added to the Science Data Server, its record will be shown in this window. The other window is called Granules and is used to display information included in the Granule database table. If a granule is inserted for an ESDT, the granule information will be listed in this window if its ESDT is highlighted in the Collection window. In addition to these two main windows, this DB Viewer GUI can also show ESDT database validation rules, Product Specific Attributes, (PSA) information, and summary information about the database reviewed.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1 The ECS Assistant has been properly installed.

2 The ECS Subsystem Manager is running.

3 The environment variables for using the database have been set correctly.

26.10.1 Starting ECS monitor GUI:

1 Invoke the ECS Assistant GUI.

- The ECS Assistant GUI will be launched.

2 At the ECS Assistant GUI, select ESDT Manager GUI by clicking the ESDT Manager.

- The ESDT manager GUI will appear.

3 At the ECS ESDT Manager GUI, select the DB Viewer by clicking the **DB Viewer** button.

- The Database Login GUI will appear as shown in Figure 26.10-1.
- Fill in the fields to point to the specific database for the mode used.
- Click Login to open the DB Viewer.
- The DB Viewer GUI will appear as shown in Figure 26.-10-2
- ESDTs are listed in the Collections window.

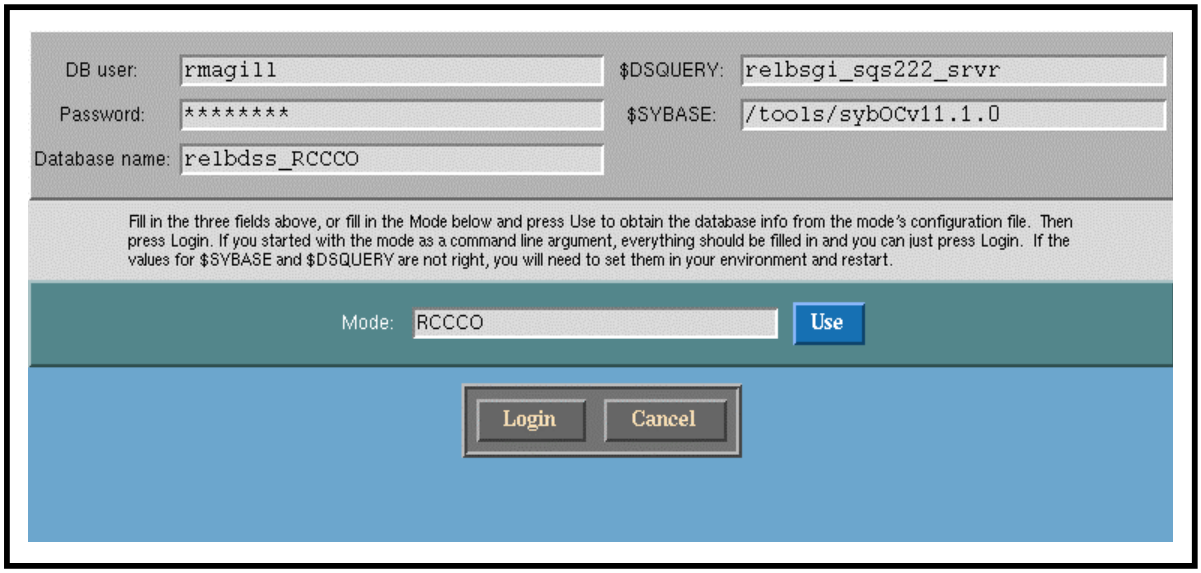


Figure 26.10-1. Database Login GUI

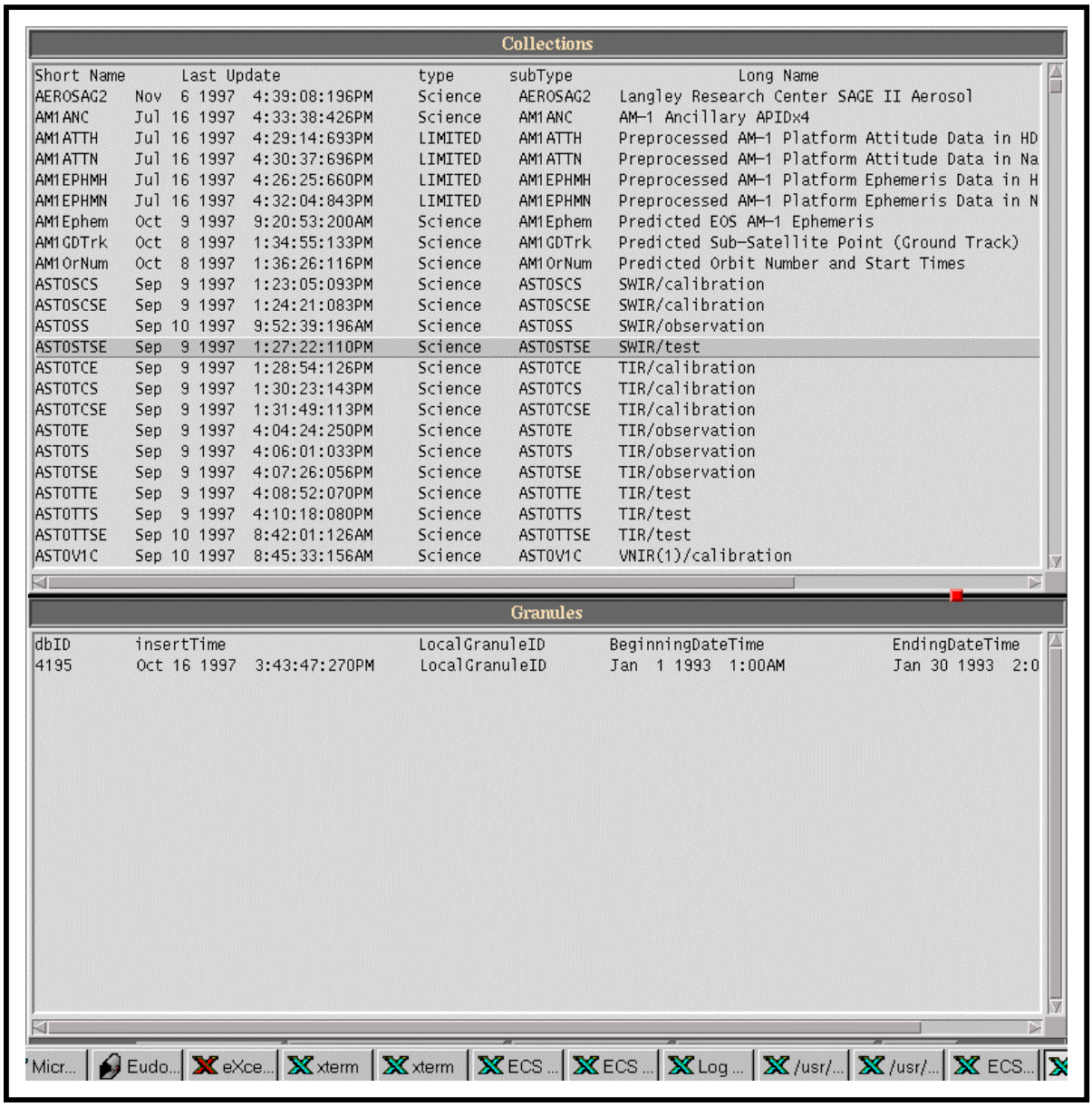


Figure 26.10-2. DB Viewer GUI

4 To view the inserted granules for a selected ESDT, first select an ESDT by clicking its short name in the Collections window.

- The selected ESDT is highlighted.
- Granule information for that ESDT, if there is any, will be listed in the Granules window.

5 To exit, click the **EXIT** button. This will end the DB Viewer GUI.

26.10.2 Using ECS Assistant to View Database

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The ECS Assistant has been properly installed.
2. The ECS Subsystem Manager is running.

To routinely start up the ECS monitor GUI, execute the procedure steps that follow:

1 At the ECS **Subsystem Manager** GUI, select a mode by **clicking** a mode in the mode list.
- The mode should be the one to be used for SSI&T.
- Once the mode is selected, the color of the subsystem name list is changed.

2 Select a subsystem by **clicking** the radio button next to the subsystem name under the subsystem component window.
- The selected subsystem radio button will be highlighted.
- The components corresponding to that the subsystem will be displayed in the component window.

3 Select a component by **clicking** a component name under the component window.
- All the servers for that selected component will be displayed in the server window.

4 **Click** the **monitor** button from the common tasks.
- This will invoke the Server Monitor GUI window as shown in Figure 26.10-3.
- The status "UP/DOWN" indicates whether the server is running.

5 To see which host each server running on, click the **cdsping all servers…** button.
- This will invoke the **cdsping GUI** as indicated in Figure 26.10.4.
- The host name for each running server is listed

6 Both **Server monitor GUI** and **cdsping GUI** can be updated by clicking the **update** button in the GUI.

3 This will cause the list to update to the current status.

7 To monitor other servers, repeat steps 1-6.

8 To exit, click the EXIT button. This will end the monitor GUI.

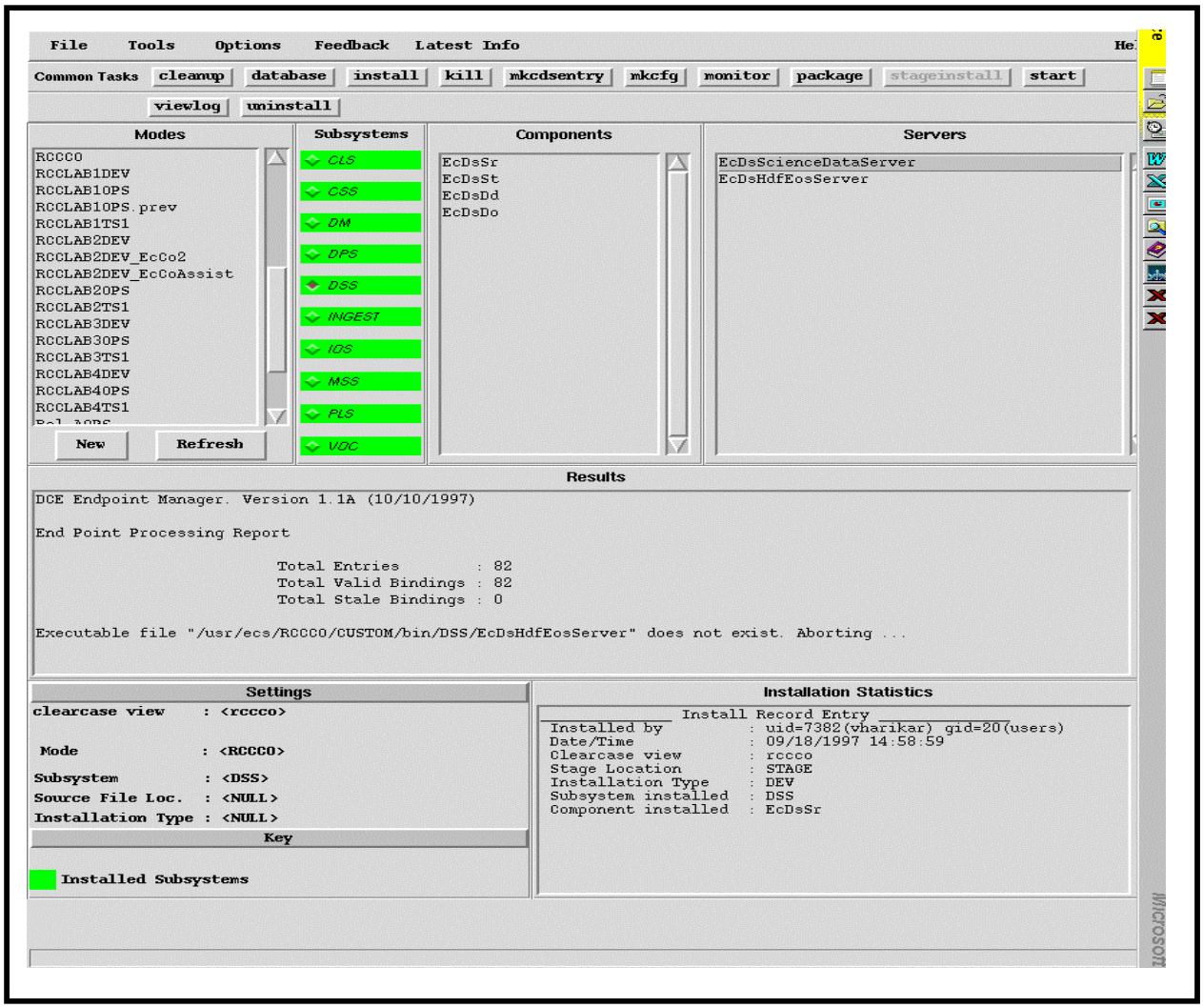


Figure 26.10-3 Server Monitor GUI

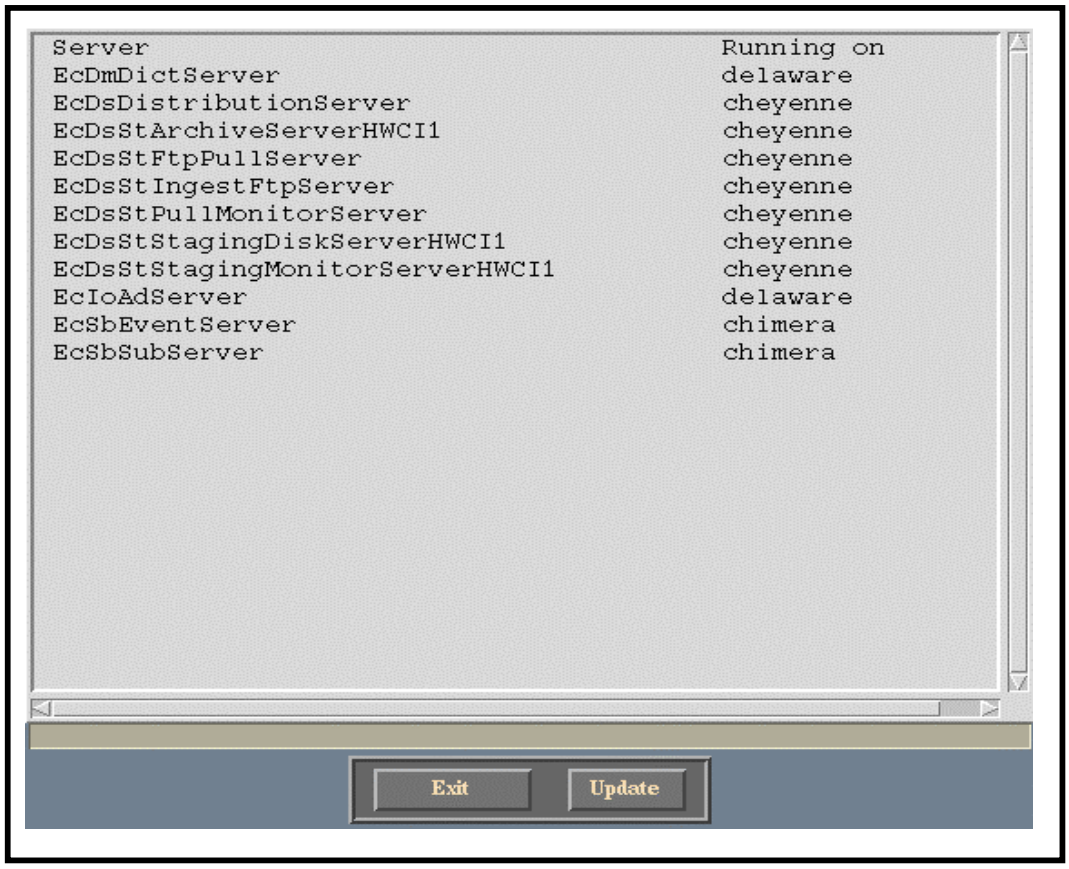


Figure 26.10-4. cdsping GUI

26.10.3 Using Browser to View ECS Science PDPS/IOS Database

Connect to the **SDSRV** database with login information as follows:

Server Name: **p0acg01_srvr**

User Name: **sdsrvApp**

Password: **welcome**

The Browser lets you view all the tables in the **SDSRV** database with the mode you have selected.

The following tables are useful to track down the problems in insert ***.met**:

1. DsDeDictionaryAttribute
2. DsMdAddtionalAttributes
3. DsMdCollections
4. DsMdGranules

26.11 Installing ESDTs and Inserting Granules on the Science Data Server

26.11.1 Required Servers for Installing ESDT's

The following servers need to be started and running before installing ESDTs (with GDAAC machine names as examples):

- **Science Data Server (g0acs03)**
- **Storage Management Servers (g0icg01, g0drg01, g0dps02)**
- **Data Distribution Servers (g0dps02)**
- **Subscription Server (g0ins01)**
- **Advertising Server (g0ins02)**
- **Data Dictionary Server (g0ins02)**

26.11.2. Installing/Removing (ESDT/DLL) using the Science Data Server Operator GUI

Before the ECS can process data, an Earth Science Data Type must be installed into the system via the Science Data Server (SDSRV). The ESDT allows the system to recognize a particular data type and also provides services for accessing the data in the form of a Dynamic Link Library (DLL). The following procedures give step-by-step instructions on configuring the ESDT and installing the ESDT using the Science Data Server GUI., see Figure 26.11.1. Science Dataserver Operator GUI.

Installing a single Earth Science Data Type (ESDT) or Dynamic Link Library (DLL)

1 Copy the ESDT descriptor file and ESDT/DLL file from the source directory to the directory under the current mode of operations, The ESDT descriptor files are installed in the specified mode.

DLL's located : /usr/ecs/<mode>/CUSTOM/lib/ESS

ESDT Descriptors Located: /usr/ecs/<mode>/CUSTOM/data/ESS

2 Ensure that the following servers are currently executing: **Advertising Service** on the appropriate ADSHW HWCI server machine, **Data Dictionary Service** on the appropriate DMGHW HWCI server machine, **Science Data Server** on the appropriate ACMHW HWCI server machine and the **Subscription Service** that operates on the appropriate CSS server machine.

3 Start the **SDSRV GUI** by entering the following at the UNIX prompt on the SDSRV GUI workstation:

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

b) login: **ID**
password:

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

d) **cd /usr/ecs/<mode>/CUSTOM/utilities/EcDsSdSrvGuiStart <mode>**

4 On the main screen select the **Data Types** tab. A list of the ESDTs that have already been installed on the SDSRV will be displayed.

5 Click the **Add** button below to bring up the **Add Data Type** window.

6 **Descriptor Filename:** enter path to where the ESDT/DLL is located, including the full ESDT descriptor. **Archive ID:** field. **Note:** The descriptor filename and DLL Filename will require the complete directory path name as part of the file name which is the same directory as was specified in step 1 above. **(isolate the particular Data Type from the larger List, by using a unique sequence of letters or numbers at the end of the full path to better identify the Data Types ie;/*___*).** To specify specific directories, the **File..** button to the right of the Descriptor Filename and DLL Filename data entry fields will bring up a standard file selection GUI for this purpose. Also note that the **Archive ID** field will be constructed using the DSS Storage Management Staging Server UR that is found in the Science Data Server configuration file. The Science Data Server Configuration file is located in:
/usr/ecs/<mode>/CUSTOM/cfg/EcDsScienceDataServer.CFG.

Example: If the **DSSSTMGSTAGEINGSERVERUR** field was set to **DRP1_OPS** then the **Archive ID** fields would be set to **DRP1_OPS.**

7 Click the **Ok** button, this will cause the **Add Data Type** window to initiate installation of the ESDT/DLL into the Science Data Server.

- The Science Data Server GUI will respond in a short time with a window stating that **MM/DD/YY HH/MM Finished adding ESDT's**.Also, the ESDT will appear alphabetically on the **Science Data Server - Data Types** list under the **Data Types** tab.

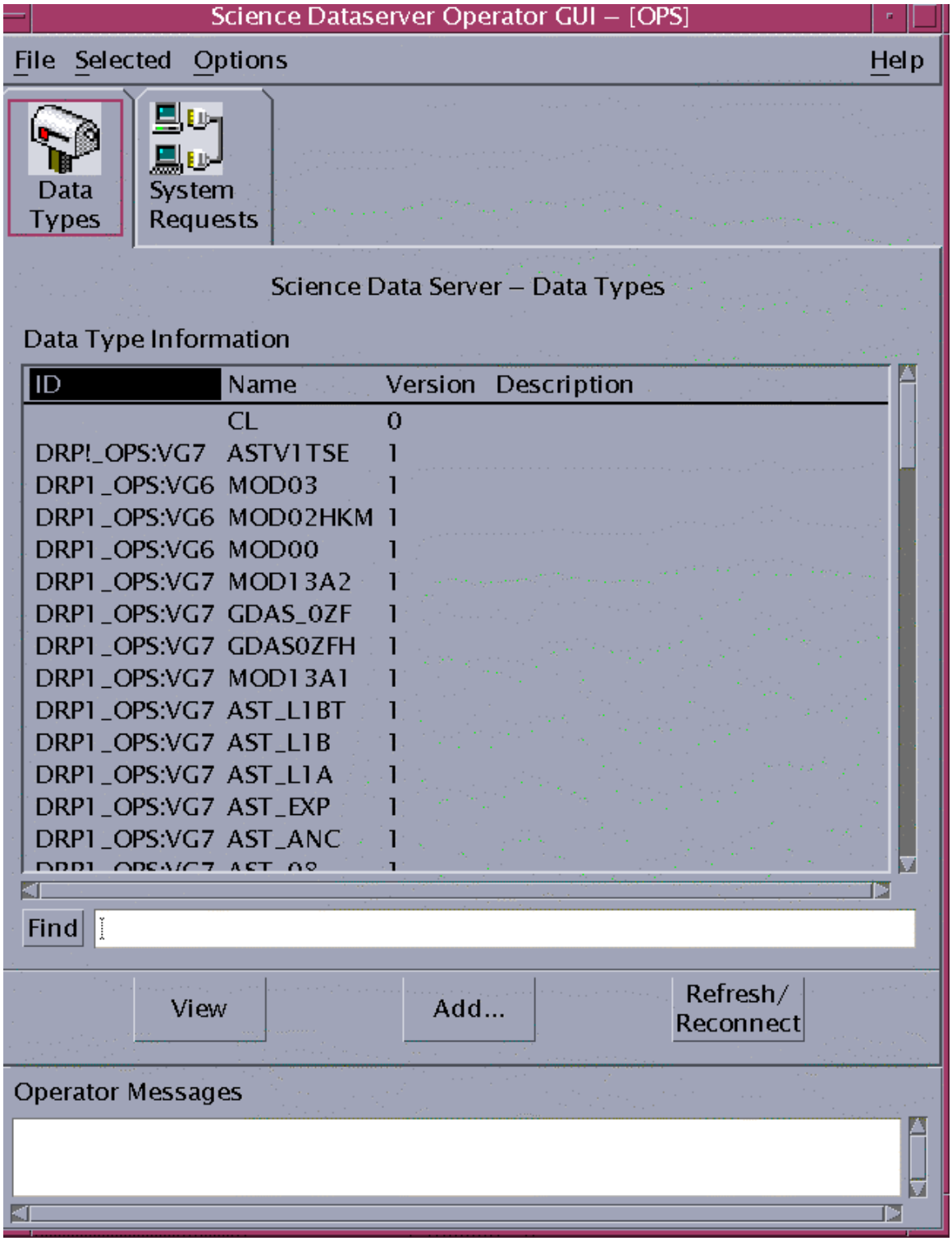


Figure 26.11.1. Science Dataserver Operator GUI.

26.11.3 Granule Insert into Science Data Server

The Science Data Server subsystem includes a utility that will allow users to manually insert a data granule into the ECS. The tool will prompt the user for key inputs for The Science Data Server subsystem includes a utility that will allow users to manually insert a data granule into the ECS. The tool will prompt the user for key inputs for inserting the granule. The following procedures describe this process.

---

1 Ensure that the **Science Data Server** subsystem is currently executing on the appropriate ACMHW HWCI server machine. Tested on **p0acs03.**

2 Start the Science Data Server test utility by entering the following at the UNIX prompt on the SDSRV workstation:

- **setenv MODE TS1**
- **cd /usr/ecs/<mode>/CUSTOM/bin/DSS/**
- **source ../ ../utilities/EcCoEnvCsh**
- **/usr/ecs/<mode>/CUSTOM/bin/DSS/dttest6 ConfigFile /usr/ecs/<mode>/CUSTOM/cfg/EcDsScienceDataServerClient.CFG**
- **ecs_mode <mode>**

3 The following selection menu is displayed:

- **INSERT granule**
- **INSERT granule with browse file**
- **ACQUIRE granule with date**
- **ACQUIRE granule with a UR**
- **DELETE granule**
- **Exit**

Please make selection=>

- Choose option **1** to perform the search and acquire.

4 The program indicates that the search process will take place first by displaying the message "**Executing insert .....** ". The user is prompted to enter the datatype of the data that will be inserted. For example:

- Enter data type=> **AST_L1B**

5 The program will then prompt the user for the full path name of the data file. For example:

- Enter datafile name (full path)=>**/tmp/:SC:AST_L1BT:1391:1.EOSHDF**

6 The program will then prompt the user for the full path name of the metadata file. For example: Enter data metafile name(full path)=>**/tmp/AST_L1BT.MCF**

7 The program will then give status on the success of the insert. The following messages should appear on the successful insert:

- **Insert science data only…**
- **Trying to make a request to {:DSSDSRV} Success.**

8 The user should hit return at this prompt and the program will redisplay the first menu that was given in step 3. The user can then choose option 6 to exit.

26.11.4 Acquire a Granule from Science Data Server

The Science Data Server subsystem includes a utility that will allow users to manually search and retrieve (acquire) a data granule from the ECS. The tool will prompt the user for key inputs for acquiring the granule. The following procedures describe this process.

Acquiring a Granule from the Science Data Server

1 Ensure that the **Science Data Server** subsystem is currently executing on the appropriate ACMHW HWCI server machine. Tested on **p0acs03.**

2 Start the Science Data Server test utility by entering the following at the UNIX prompt on the SDSRV workstation:

- **setenv MODE TS1**
- **cd /usr/ecs/<mode>/CUSTOM/bin/DSS/**
- **source ../ ../utilities/EcCoEnvCsh**
- **/usr/ecs/<mode>/CUSTOM/bin/DSS/dttest6 ConfigFile /usr/ecs/<mode>/CUSTOM/cfg/EcDsScienceDataServerClient.CFG**
- **ecs_mode <mode>**

3 The following selection menu is displayed:

- **INSERT granule**
- **INSERT granule with browse file**
- **ACQUIRE granule with date**
- **ACQUIRE granule with a UR**
- **DELETE granule**
- **Exit**

Please make selection=>

- Choose option **3** to perform the search and acquire.

4 The program indicates that the search process will take place first by displaying the message "**Executing search .....** ". The user is prompted to enter a hostname. Enter the hostname of the machine on which the Science Data Server process is executing:

- Enter hostname=> **dss2**

5 The user is prompted to enter a data type. Enter the ESDT short name of the type of data that is to be acquired. An example:

- Enter data type=> **AST_L1BT**

6 The user is prompted to enter a start and end date. These dates indicate a range over which the user would like to search the database for data of the given type. The start and end dates will narrow the search to those data granules that were collected within that time range. Times are given in the format **mm/dd/yy**. For example:

- Enter starting date(mm/dd/yy)=> **07/04/97**
- Enter end date(mm/dd/yy)=> **07/05/97**

7 After the start and end dates are entered the utility will make a request of the Science Data Server and the following message will be displayed:

- **Trying to make a request to [:DSSDSRV]**

A table of data granules will be displayed to the user if any were found within the given time range. For example, if the user had requested to display all of the data granules for

the ESDT **AST_L1BT** within a certain time range, the following table would be displayed:

- **UR Type Create Date Size**
- **AST_L1BT SC:AST_L1BT:1390**
- **AST_L1BT SC:AST_L1BT:1391**
- **AST_L1BT SC:AST_L1BT:1322**
- **AST_L1BT SC:AST_L1BT:1289**
- **AST_L1BT SC:AST_L1BT:1299**

The table displays the data type (AST_L1BT) and UR (i.e., SC:AST_L1BT:1390) for each granule found.

In addition to the table, a list of the granules with a corresponding numerical index will be displayed with a prompt to the user to enter the index of the granule that they wish to acquire. An example of the indexed list follows:

NOW ENTERING ACQUIRE
There is(are)5 in the collection
Index = 0 AST_L1BT SC:AST_L1BT:1390 Index = 1 AST_L1BT SC:AST_L1BT:1391 Index = 2 AST_L1BT SC:AST_L1BT:1322 Index = 3 AST_L1BT SC:AST_L1BT:1289 Index = 4 AST_L1BT SC:AST_L1BT:1299
Please enter the index of the associated UR

8 At this time the user should enter the numerical index of the granule that they wish to acquire and hit enter.

9 The SDSRV utility will prompt the user for a media type, the type of media on which the data will be retrieved:

- **Valid Media Types:**
- **FtpPull**
- **FtpPush**
- **8MM**
- **4MM**
- **CDROM**
- **9TRK**
- **Enter media type (case sensitive)=>**

The user should enter one of the values of the valid media types. If the user entered FtpPush, the user will expect the data to be ftp'd to a given directory.

10 After the user has entered the Media type, the utility will prompt the user for the media format. The media format should be entered as "FILEFORMAT"

Enter mediaformat=> **FILEFORMAT**

11 After the user has entered the Media format, the utility will prompt the user for the user profile id. This entry can be any alphanumeric character.

Enter userProfID => **a**

12 After the user has entered the user profile id, the utility will prompt the user for a user id and associated password. The user id/password will be used for authorization to perform the ftp of the data file from the archive area to the user-specified directory. The password field will not be echoed to the screen. The following is only an example. The user should use a valid user id/password within the current environment.

Enter username=> **sdsrv**
Enter password=>

13 The next entry that the user must enter is the host id of the machine to which they want the data ftp'd. Enter a valid host name as follows:

Enter host=> **dss2**

14 After the host id, the user must enter the fully qualified destination directory to which the file will be ftp'd:

Enter destination=> **/tmp**

15 After the destination directory has been entered, the utility will give the following message:

Trying to make a request to [:DSSDSRV]

If the acquire operation is successful, the utility will give the following message:

Acquire successful.

Please <CR> to continue.

16 The user should hit return at this prompt and the program will redisplay the first menu that was given in step 3. The user can then choose option 5 to exit.

26.11.5 Removing ESDTs using the Command Line:

Procedures:

1 **telnet** to (SDSRV) **p0acs03**[e.g.]
2 login: **id**, **password**:
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:
****
6 **rlogin p0ins02-l cmts1**
7 *Login to DCE* **(dce_login <name> <Password>), *setenv DISPLAY ….:0.0***
8 **rlogin p0ins02 -l ios**
1. *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

14 your_short_name_here

- y

Success is when the "<" prompt returns
To make sure the advertisements are deleted from the database

15 incagold% isql -Uios_role -Pwelcome –**Sp0ins02**_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.
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!
****

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

26.12 Production Planning Considerations

1 During normal operations it is expected that the Production Planner will not have to add PRs to the PDPS database very frequently. The frequency of this activity is, to some extent, determined by the SCF responsible for the science software.

- The PR is a template request to generate a particular data product and results in a production run of the associated SCF-provided PGE.
- PR specifies a range (temporal, orbit, or tile) over which the data products are to be produced or the PGEs are to be scheduled.
- PR might request that the data product be produced for only a single day's data.
- PR might request that data products be produced for every opportunity of input data for several months, resulting in several hundred jobs being planned and run as the input data become available.
- Early in a mission the SCF may prefer to request processing for a short time period only (e.g., a week or less).
- At that time the SCF is gaining an understanding of the on-orbit behavior of the instrument, the resulting data, and the interaction of the science processing software with real data.
- SCF reviews the quality of the products and notifies the Production Planner of the need for any changes to the PR (e.g., discontinue the PR, change time ranges, or modify input parameters).
- When the SCF has developed a good understanding of the instrument's behavior, the team may be comfortable requesting processing for months at a time.
- DAAC operations may have operational reasons for wanting to issue processing requests for a more limited time period.

2 The Production Planner has to balance the various considerations when determining whether or not to create or update a PR.

Planning decisions are made on the basis of locally defined planning strategies for supporting the SCFs' data processing needs. The production planning tools are intended to be flexible enough in their design to support the particular planning and scheduling cycles of the operations organization at each DAAC.

Before planning production the Production Planner must coordinate with the Resource Planner to resolve all resource allocation issues. The Resource Planner notifies the Production Planner of the resources available for use in processing. Furthermore, the Production Planner may well have direct access to the Resource Plan.

The Production Planner prepares monthly and weekly production plans. In addition, the Production Planner develops a daily production schedule from the most current weekly plan. However, the first step in the planning process is creating production requests using the Production Request Editor.

26.12.1 DPREP Considerations

DPREP (data preprocessing) is a set of three PGEs that are supplied by ECS, unlike most PGEs, which are provided by the Science Computing Facilities that ECS supports. DPREP consists of the following three PGEs:

- EcDpPrAm1EdosEphAttDPREP_PGE (Step 1).
- EcDpPrAm1FddAttitudeDPREP_PGE (Step 2).
- EcDpPrAm1FddEphemerisDPREP_PGE (Step 3).

The PGEs run separately and in a particular sequence.
Three files describe the PGEs and how to run them:

- "DPREP_README"
- "HowToCreateDprepTarFile"
- "HowtoRunDPREP"

The files are installed on the science processor hosts (e.g., e0spg01, g0spg01, l0spg01, n0spg03) in the **/usr/ecs/*MODE*/CUSTOM/data/DPS** directory.
The DPREP PGEs process Level Zero (L0) Terra (AM-1) spacecraft data (e.g., ESDT AM1ANC) provided by EDOS. The output files/granules of the DPREP PGEs are subsequently used in the processing of data from various instruments on the satellite. They provide the following types of ancillary (non-science) data:

- Ephemeris
- Spacecraft location: ephemeris (or orbit) data include: latitude, longitude, and height.
- Attitude
- Orientation of the satellite, including yaw, pitch, and roll angles; and angular rates about three axes.
- There are two profiles for DPREP PGEs:
- Profile 1 runs routinely at the DAACs using previous DPREP output in addition to new Terra ancillary (e.g., AM1ANC) data.
- Profile 2 (the boot-up procedure) takes in the Terra ancillary data only and is run under two sets of conditions:
- First run of DPREP (because there is no previous output) to initialize DPREP processing.
- Following any long period of time during which EDOS L0 ancillary data are unavailable. (Short gaps in the ephemeris data are filled by EcDpPrAm1EdosEphemerisRepair, one of the executables in the EcDpPrAm1EdosEphAttDPREP_PGE.)

In order to run Profile 2 successfully following a long period of data unavailability, DPREP must be told where to resume orbit counting. The initial orbit number in the Step 1 process control file (PCF), must be set to the orbit number corresponding to the timestamp at which data availability resumes.
Until an automated process can be implemented, whenever there is a telemetry drop-out, a member of the DAAC science support team takes the following actions:

- Calls the Flight Operations Team (FOT).
- Asks for the on-line engineer.

- Requests the orbit number that coincides with the start time of the first L0 ancillary data set that follows the data drop-out.
- Sets the orbit number in the Step 1 PCF.

Then Profile 2 can be run successfully. Afterward, routine operations can be resumed using Profile 1 PGEs.

This page intentionally left blank.

26.13 PGE Registration and Test Data Preparation

The integration of science software with ECS requires that information about the Product Generation Executives (PGEs) be made known to the PDPS in its database. In addition, the PGEs themselves and the test files that they use (both input and output) need to be placed on the Data Server. These steps must be accomplished before the science software can be run and tested within the ECS.

The following procedures describe how to register a new PGE with ECS. This involves updating the PDPS database with information needed to plan, schedule, and run the PGE. The first step in the PGE registration process is to determine which ESDTs are needed for the PGE. You must Verify that an ESDT metadata ODL file exists for each ESDT or generate an ODL file. The next step in the process is to create a PGE metadata ODL file using the delivered PCF. Finally, additional operational information (resource requirements and runtime statistics) must be input into the PDPS database. This is the last step in the PGE registration process. The order in which these procedures are done is important and should be done as indicated. Please reference Appendix C. for Examples of PGE and ESDT ODL Files for Each Instrument Team**.**

26.13.1 PGE ODL Preparation

. **This section describes how to prepare PGE ODL files. It is assumed that the SSIT Manager is running** .

---

1 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **PCF ODL Template**.

- An xterm with title "SSIT: Science Metadata ODL Template Creation" will be displayed.

2 At the program prompt **Configuration Filename (enter for default: ../../cfg/EcDpAtCreatODLTemplate.CFG)?**

- Press **Enter** for the default configuration file

3 At the program prompt **ECS mode of operations?,** type ***mode***, press **Return** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **OPS** or **TS1**.

4 At the program prompt **Process Control file name (PCF to generate template from)?**, type ***PCFpathname*/*PCFfilename***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the PCF. If not specified, the directory from which the SSIT Manager was run will be assumed.
- The ***PCFfilename*** is the file name of the PCF.

5 At the program prompt **PGE name (max 10 characters)?**, type ***PGEname***, press **Return**.

- The ***PGEname*** is the name of the PGE that will be registered.

6 At the program prompt **PGE version (max 10 characters)?**, type ***PGEversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***PGEversion*** is the version of the PGE that will be registered.

7 At the prompt **PGE Profile ID (0 for Null, max 999)?,** type 1 or any valid profile ID.

- After a brief time, the message “Successfully created ODL template file” should be displayed if the task was successful.
- The program will output a file with the filename **PGE_*PGEname#PGEversion#ProfileID*.tpl**.
- For example, if the PGE name was **PGE35**, and the version and profile ID were both **1** this output file will be named **PGE_PGE35#001#01.tpl**.

8 At the program prompt **Hit return to run again, ‘q <return>’ to quit:**, press **Return** to repeat process with another PCF or type **q** and press **Return** to quit.

- The xterm will disappear.

9 At a UNIX prompt on an AIT Sun, type **cd *SSITrunPathname***, press **Return.**

- The ***SSITrunPathname*** is the full path to the directory from which the SSIT Manager was run, for example /usr/ecs/TS1/CUSTOM/bin/DPS. This will be the directory where the file **PGE_*PGEname#PGEversion#ProfileID*.tpl** will reside.

10 At a UNIX prompt on the AIT Sun, type **cp PGE_*PGEname#PGEversion#ProfileID*.tpl PGE_*PGEname#PGEversion#ProfileID*.odl**, press **Return.**

- The **PGE_*PGEname#PGEversion#ProfileID*.tpl** is the file name of the ODL template file created in step 7.
- The **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of a copy which can be safely edited. This file name convention must be used.

11 At a UNIX prompt on the AIT Sun, type **mv PGE_*PGEname#PGEversion#ProfileID*.odl /usr/ecs/<mode>/CUSTOM/data/DPS/ODL**

- This will place the ODL file in the directory where the executable that populates the PDPS database will read from. **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of the copy created in step 10.

12 At a UNIX prompt on the AIT Sun, change the directory to the one in step above and type **vi PGE_*PGEname#PGEversion#ProfileID*.odl**, press **Return.**

- The **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of the copy created in step 10.
- Any text editor may be used such as *emacs*. For ex**ample, emacs PGE_PGE35#001#01.odl, press Return.**

13 In the file, add required metadata to the ODL template.

- For an explanation of what metadata is required, see file /usr/ecs/<mode>/CUSTOM/data/DPS/PGE_ODL.template.
- Note that the ShortNames typed into this file must each have a corresponding PDPS ESDT metadata ODL file (sec. 26.13.2).
- All objects corresponding to output ESDTs will automatically have the SCIENCE_GROUP and YIELD set during the generation of PGE ODL.
- All objects corresponding to output ESDTs will have an attribute “ASSOCIATED_MCF_ID. Place here the Logical Unit Number (LUN) listed in the PCF for the associated MCF listing.
- All objects corresponding to static input ESDTs must have the SCIENCE_GROUP set. Objects corresponding to *dynamic* input ESDTs should NOT have the SCIENCE_GROUP set.
- See Appendix E for an example of PCF and corresponding PGE ODL files.

14 Save the changes made to the ODL template file and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.
- For other editors, refer to that editor's documentation.

4. If you make a mistake entering any values, press **Return** here; your previous enteries are restored as defaults and you won't have to retype them.
5. A comment should be received: "**Update of PDPS/SSIT database with PDPS SCIENCE METADATA SUCCESSFUL"**

26.13.2 ESDT ODL Preparation

Assumption:

The PGE ODL file has been created and edited for the required PGE.

Follow the steps below to prepare ESDT ODL files for each ESDT required by the PGE.

---

1 Determine ShortName for required ESDTs corresponding to a Logical Unit Number (LUN) in the PGE ODL file.

2 At a UNIX prompt on an AIT Sun, type **ls /usr/ecs/<mode>/CUSTOM/data/DPS/ODL/ESDT_*ShortName#Version*.odl**, press **Return**.

- The **ESDT_*ShortName#Version*.odl** is the file name of the ESDT ODL file you are looking for where ***ShortName*** is the ESDT's ShortName and ***Version*** is the ESDT version. If a file for the desired ESDT is listed, then it has already been prepared and this procedure can be exited now.
- For example, if the desired ESDT has the ShortName MOD03 and version 001, type **ls /usr/ecs/TS1/S/CUSTOM/data/DPS/ODL/ESDT_MOD03#001.odl,** press **Return**.
- If the desired file is *not* listed, continue on to step 3.

3 At a UNIX prompt on the AIT Sun, type **cd *WorkingPathname***, press **Return.**

- The ***WorkingPathname*** is the full path name to a working directory for which the user has write permissions.
- For example, **cd /home/jdoe/working/**, press **Return**.

4 At a UNIX prompt on the AIT Sun, type **cp /usr/ecs/<mode>/CUSTOM/data/DPS/ESDT_ODL.template ESDT_*ShortName#Version*.odl**, press **Return**.

- For <mode> enter the mode you are working in, for example **OPS** or **TS1**.
- The **ESDT_*ShortName#Version*.odl** is the file name of the ESDT ODL file to be created.
- This command copies a template ESDT ODL file to the ESDT ODL file to be created. The template is well commented.

- For example, type **cp /usr/ecs/<mode>/CUSTOM/data/DPS/ESDT_ODL.template ESDT_MOD03#001.odl,** press **Return**.
- The **ESDT_*ShortName#Version*.odl** file naming convention *must* be observed.

5 At a UNIX prompt on the AIT Sun, type **vi ESDT_*ShortName#Version*.odl**, press **Return**.

- The **ESDT_*ShortName#Version*.odl** represents the file name of the ESDT ODL template file created in step 4.
- Any text editor may be used such as *emacs*. For example, **emacs ESDT_MOD03#001.odl**, press **Return**.

6 In the file, add required metadata to the ODL template.

- Use the internal documentation contained in the ODL file (from the original template) to aid in populating with metadata.
- Note that the ShortName specified within the file must match the ShortName of the file name itself.
- In addition, the ShortNames used in the PDPS PGE metadata ODL file must match the ShortNames in these files.

7 Save the changes made to the ESDT metadata ODL file and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.
- For other editors, refer to that editor's documentation.

8 Next type **mv ESDT_*ShortName#Version.odl* /usr/ecs/<mode>/CUSTOM/data/DPS/ODL.**

- This will place the just created ESDT ODL file in the directory where PDPS will read it from.

9 Repeat steps 1 through 8 for each ESDT required by a particular PGE. When all ESDT metadata ODL files have been completed, continue on to next section.

26.13.3 Update PDPS/SSIT Database with PGE Science Metadata

In order to update the PDPS Database with PGE metadata, the ESDT metadata ODL files must first be prepared for each ESDT required by the PGE. This section describes how to perform the next step, running the SSIT Science Update program.

Assumptions:

1. The SSIT Manager is running.
2. The directory used for containing the PDPS PGE metadata ODL files. Nominally, this is **/usr/ecs/<mode>/CUSTOM/data/DPS/ODL.**

Updating the PDPS Database with PGE Metadata

The following is a list of tools, procedures and or assumptions:

The directory used for containing the PDPS ESDT metadata ODL files can be accessed by the following commands:

---

1 telnet to (AITTL/DPS) **p0ais01** or a machine that matches the SSIT Manager host.

2 login: **ID**, password:

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

4 The directory used for containing the PDPS ESDT metadata ODL files. is **/usr/ecs/<mode>/CUSTOM/data/DPS/ODL**

5 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **SSIT Science Metadata Update**.

- An xterm with title "SSIT: Science Metadata Database Update" will be displayed.

6 At the program prompt **Configuration Filename (enter for default: ../../cfg/EcDpAtRegisterPGE.CFG)?**

- Press **Return**.

7 At the program prompt **ECS mode of operation?**, type ***mode***, press **Return** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **OPS** or **TS1.**

8 At the program prompt **PGE name (max 10 characters)?**, type ***PGEname***, press **Return**.

- The ***PGEname*** is the name of the PGE that will be registered. This name must match the PGE name specified.

9 At the program prompt **PGE version (max 10 characters1)?**, type ***PGEversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***PGEversion*** is the version of the PGE that will be registered. This version must match the PGE version specified.

10 At the program prompt **PGE Profile ID (0-999, 0 means null)?** Type in a valid profile ID and press **Return,** or if already listed just press **Return.**

- The PDPS database will then be updated with the information contained in the file **PGE*_PGEname#PGEversion#ProfileID*.odl**

11 At the program prompt **Hit return to run again, q <return> to quit:**, press **Return** to update the PDPS database with another PGE ODL metadata file or type **q** and press **Return** to quit.

- If you make a mistake entering any values, press **Return** here; your previous entries are restored as defaults and you won't have to retype them.
- NOTE: If you make mistakes while editing the PGE and ESDT ODL files, you can run the ODL checker (Tools → PDPS Database → Check ODL) via the SSIT manager to locate any errors.

ODL files must have been created to define the PGE to PDPS. Examples of the ODL files are under the data directory: PGE_ODL.template, ESDT_ODL.template, ORBIT_ODL.template, TILE_ODL.template and PATHMAP_ODL. A tool can be run to generate a template ODL file for the PGE from the SSIT Manager via Tools->PDPS Database->PCF Odl Template script. This then has to be populated with all information that can not be garnered from the PCF. The CheckOdl tool from the SSIT Manager via Tools->PDPS Database-

>Check ODL can be used to flag any errors in ODL before trying to put it in the database.

Sample of ESDT.odl files being established in ECS

home/emcleod/MODIS/STORE/PGE07
p0ais01{emcleod}10: ls
ESDT_MD10L2#001.odl MOD_PR10 pge_cfg
ESDT_MD35L2#001.odl MOD_PR10.mk scf_cfg
ESDT_MOD02H#001.odl PGE07.mk script
ESDT_MOD03#001.odl doc
p0ais01{emcleod}11: cp ESDT_MD10L2#001.odl ESDT_MD35L2#001.odl
ESDT_MOD02H#001.odl ESDT_MOD03#001.odl /usr/ecs/OPS/CUSTOM/data/DPS/ODL/

26.13.3.1 AlternativeTool for SSIT Metadata Update:

Source the buildrc file for the mode in which you are working (*source .buildrc)*.
/usr/ecs/<MODE>/CUSTOM/utilities, Note that this only has to be done once per login. Then **(*cd* /usr/ecs/<MODE>/CUSTOM/bin/DPS)**

(The tool can also be executed by being in the /**usr/ecs/<MODE>/CUSTOM/bin/DPS** and executing **EcDpAtDefinePGE..)**

Shell script prompts user for information.

1 Enter in the location of the configuration file **(../../cfg/EcDpAtRegisterPGE.CFG).**
2 Filename Enter the MODE of operation (<MODE>).
3 Enter name of PGE (it must match what is in the PGE ODL file).
4 Enter the version of the PGE (it must match what is in the PGE ODL file).
5 Enter the Profile ID (it must match what is in the PGE ODL file). Note that the ODL file for the PGE must have the of: PGE_<PGE NAME>#<PGE VERSION>#<PROFILE ID>.

- Each ODL file is displayed as it is processed. A good status message should be displayed as a result. Information about the PGE (inputs and outputs, Production Rules, etc) should be entered in the Database.

26.13.3.2 Examples of PGE and ESDT ODL Files for Each Instrument Team

This section is taken from the latest **Green Book 162-TD-001-005** and are listed in Appendix C. Depicted are examples of ODL files in SSI&T activities. Then, examples of specific ODL files are listed by instrument (ASTER, MISR or MODIS).

Template ODL Files

There are five Template ODL files listed therein. The specific or tailored ODL files listed were derived from these templates by appropriate editing and filling-in of values. The three ODL Template files listed reside, on the AIT Sun host, at **/usr/ecs/<mode>/CUSTOM/data/DPS** . They are

PGE_ODL.template
ESDT_ODL.template
ORBIT_ODL.template
PATHMAP_ODL.template
TILE_ODL.template

Example of a successful PDPS Science Metadata Update:

```
PDPS/SSIT SCIENCE Metadata Database Update **
Configuration filename?  (enter for default: ../../cfg/EcDpAtRegisterPGE.CFG)
ECS Mode of operations?  (enter for default: OPS)
OPS
PGE name (max 10 characters)?
PGE07
PGE version (max 10 characters)?
001
PGE Profile ID (0-999, 0 means null)? (enter for default: 1)
1
Warning: Could not open message catalog "oodce.cat"
EcDpAtRegisterPGE: Process Framework: ConfigFile
../../cfg/EcDpAtRegisterPGE.CFG  ecs_mode OPS
'  PGE profile id = '1' ...
Do you wish to overwrite the previous PGE PGE07( (y)es or (n)o):
y
FILES PROCESSED
:  PGE SCIENCE ODL file =/usr/ecs//OPS/CUSTOM/data/DPS/ODL/PGE_PGE07#0#001.odl
   ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MOD02H#001.odl
   ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MD35L2#001.odl
   ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MOD03#001.odl
   ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MD10L2#001.odl
********** Update of PDPS/SSIT database with  PDPS SCIENCE metadata SUCCESSFUL **********
Hit return to run again, 'q <return>' to quit:
```

26.13.4 Operational Metadata

The SSIT version of the PDPS database is initialized and updated with SSIT Operational Metadata so that the Planning and Processing Subsystem can schedule and run PGEs. Here, PDPS Operational Metadata refers to PGE information which is supplied to the DAAC/SSIT Operator and may change frequently.

The operator enters this data directly into the SSIT Operational Metadata Update GUI. The program then writes the data directly to the SSIT version of the PDPS database.

Before running the SSIT Operational Metadata Update from the SSIT Manager, you must first update the PDPS with SSIT Science Metadata. In addition, to get initial PGE Performance data which will be entered into the GUI, you need to run the profiling utility, EcDpPrRusage on the PGE or have the information on profiling provided. See section 26.13.2.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The required UNIX environment variables have been set.

2. The Science metadata has been updated to the PDPS database for this PGE.

To update the SSIT version of the PDPS database with operational metadata, execute the steps that follow:

---

1 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **SSIT Opnl Metadata Update**.

- The PDPS/SSIT Database Update GUI will be displayed.

2 Click on the radio button labeled **NEW PGE** in the lower left quadrant.

- The PGE that you are working on should appear in the subwindow labeled **PGE Names** along with its version number in the subwindow labeled **PGE Versions**.

3 In the subwindow labeled **PGE Names**, click on a PGE name. Then in the subwindow labeled **PGE Versions**, click on the PGE version for that PGE. Then click on the button labeled **EDIT**.

- The PGE name and version will be highlighted when you click on them.
- The page tabs **PROFILE**, **RUNTIME**, and **ESDT** will change from gray (indicating disabled) to black (indicating enabled).
- To see the contents of PGE Metadata, click on the button labeled **DISPLAY** and then click on the button labeled **DONE**.
- If the PGE name and/or version does not appear in the lists, it means that updating of PDPS database with PGE metadata was not successful.

4 Click on the **PROFILE** page tab.

- The Profile page will be displayed.

5 In the fields under the label **Performance Statistics**, enter the information specified.

- In the field labeled **Wall clock time**, enter the amount of wall clock time it takes for one execution of the PGE, in seconds. The tab **PROFILE** will change from black (indicating enabled) to red (indicating database needs to be updated by APPLY button).
- In the field labeled **CPU time (user)**, enter the so-called *user* time of the PGE, in seconds. This value should come from profiling the PGE .
- In the field labeled **Max memory used**, enter the maximum amount of memory used by the PGE, in megabytes (MB). This value should come from profiling the PGE .
- In the fields labeled **Block input ops** and **Block output ops**, enter the integer number of block inputs and block outputs, respectively. These values should come from profiling the PGE .
- In the field labeled **Swaps**, enter the integer number of page swaps from the PGE. This value should come from profiling the PGE .
- In the field labeled **Page faults**, enter the integer number of page faults from the PGE. This value should come from profiling the PGE .

6 In the fields under the label **Resource Requirements**, enter the information specified.

- In the field labeled **DISK SPACE used for PGE run**, enter the maximum amount of disk used by the PGE during execution, in megabytes (MB). Space should be allowed for the executable(s), input files, output files, ancillary files, static files, MCFs, and the PCF. (This number should also be in the PGE metadata ODL file; yes, there is duplication here.)
- Click on the radio button labeled **Proc. String** (if not already clicked on).

- A list of processing strings should appear in the scrollable window to the left of the two radio buttons **Proc. String** and **Computer Name**. Nominally, only one item should be listed and should be highlighted.
- In the field labeled **Number of CPUs**, the number 1 should appear.

7 Once the fields on the **PROFILE** page have been completed, click on the **APPLY** button.

- This will update the PDPS database with the information just entered. The tab **PROFILE** will change from red (indicating database needs to be updated) to black (indicating enabled).
- An information box will be displayed; click on **Ok**.
- To start over, click on the **RESET** button. This will clear all fields.

8 Click on the **File** menu and select **Exit**.

- This will end the session with PDPS/SSIT Database Update and the GUI will disappear.

4

26.13.5 SSIT Operational Metadata Update GUI

The SSIT version of the PDPS database is initialized and updated with SSIT Operational Metadata so that the Planning and Processing Subsystem can schedule and run PGEs. Here, PDPS Operational Metadata refers to PGE information which is supplied to the DAAC/SSIT Operator and may change frequently.

The operator enters this data directly into the SSIT Operational Metadata Update GUI (Figure 26.13.5-1). The program then writes the data directly to the SSIT version of the PDPS database. The SSIT Operational Metadata Update GUI is used to view or update the following operational parameters for a particular PGE:

- Performance parameters for the PGEs.
- Resource parameters for the PGEs.
- PGE user-defined static parameter.
- View the PGE science metadata file.

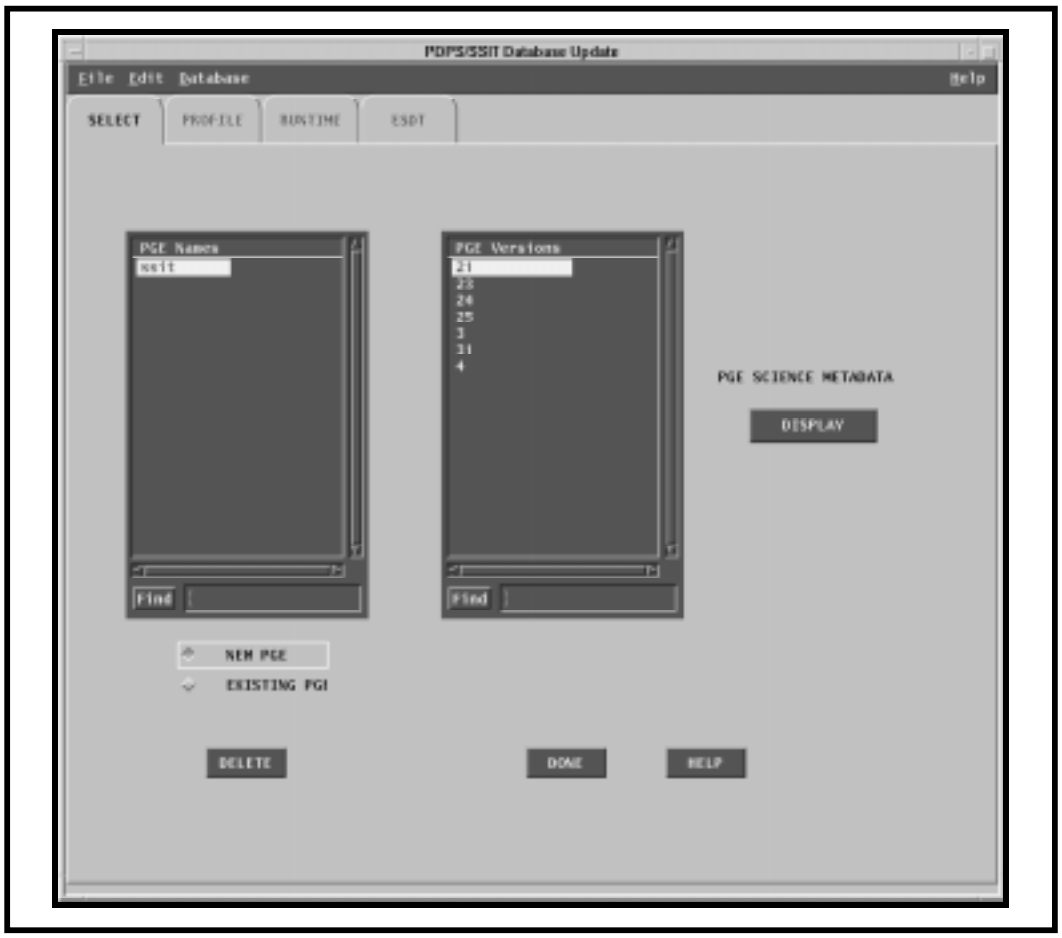


Figure 26.13.5-1. SSIT Database Operational Metadata Update GUI – SELECT view

26.13.6 Test Data Preparation and Insertion of Data Granules

This section describes how to prepare test data for use by registered PGEs. When PGEs are first delivered to the DAAC and registered within the PDPS, they will typically be run in isolation. That is, they will be run without any PGE dependencies. For this testing to be possible, test input data granules required by the PGE need to be pre-Inserted to the Data Server.

Data granules can be *dynamic* or *static*. Dynamic data granules are those whose temporal locality differs for each instance of the granule. Examples of dynamic granules are Level 0, Level 1, and Level 2 data sets. Static data granules are those whose temporal locality is static over long periods of time. Examples of static granules are calibration files which may only change with a new version of a PGE. For any granule to be Inserted to the Data Server, a Target MCF is needed (also known as an ASCII metadata ODL file or a **.**met file).

In the actual production environment, a Target MCF is produced by the PGE during execution. Thus, the data granule can be Inserted. In isolation testing of a PGE, however, the inputs needed by it will not have been Inserted by a previous PGE in the chain. This Insertion must be done

manually. The next two sections describes how to use the Source MCF for a dynamic data granule to create a Target MCF. and then describes how to do the Insert. In this way, a dynamic data granule can be Inserted to the Data Server as if a PGE had produced it.

26.13.6.1 Generating a Metadata Configuration File (Source MCF)

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

- The SSIT Manager is running.
- ESDT's are installed onto the **Science Data Server.**

To Generate the Metadata Configuration File (Source MCF) for the input and output ESDT's, execute the steps that follow.

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Get MCF.**

- An xterm in which EcDpAtGetMCFis running will be displayed as SSIT: Acquire MCF..
- Alternatively, the same tool can be invoked by typing at a UNIX prompt on an AIT Sun **EcDpAtGetMCF.sh**, press **Return**.

2 At the program prompt **Configuration Filename (default *defaultConfigFile*)?**

- Type in **../../cfg/** ***defaultConfigFile*** and press **Return**.
- The ***defaultConfigFile*** will be replaced by the full path name and file name of the default configuration file. The file name will be **EcDpAtGetMCF.CFG** where *daac* will be replaced by one of {GSFC, EDC, LARC, NSIDC}.

3 At the program prompt **ECS mode of operation (enter for default: def*aultMode*)?**, type ***mode***, press **Return** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **TS1**.

4 At the program prompt **ESDT Short Name?**, type ***ESDT ShortName***, press **Return**.

- The ***ESDTShortName*** is the name of the ESDT that the EcDpAtGetMCF tool will use to generate the MCF.

5 At the program prompt **ESDT Version?**, type ***ESDTversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***ESDTversion*** is the version of the ESDT.

6 At the program prompt **Directory to receive MCF (must be full path)?**, type ***MCFpathname***, press **Return**.

- The ***MCFpathname*** is the full path name to the location where the source MCF will be placed. For example, /home/jdoe/ssit.

7 To the final prompt **Hit return to run again, 'q <return> to quit:**, press **Return** to generate another Source MCF or type **q** and press **Return** to quit.

- ↑ If you make a mistake entering any values, press **Return** here; your previous entries are restored as defaults and you won't have to retype them.

Example of a successful installation of a Source MCF:

Configuration filename? (enter for default: ../../cfg/EcDpAtGetMCF.CFG)
ECS Mode of operations?
OPS
ESDT Short Name?
MOD03EM
ESDT Version?
0
Directory to receive MCF? (must be full path)
/home/emcleod/MCF/
Warning: Could not open message catalog "oodce.cat"
EcDpAtGetMCF: Process Framework: ConfigFile ../../cfg/EcDpAtGetMCF.CFG
ecs_mode
OPS
incomplete group entries in the configfile,using default G1
Request for MCF successful for:
ESDT name = 'MOD03EM'
ESDT version = '0'
directory = '/home/emcleod/MCF/'
Hit return to run again, 'q <return>' to quit:

26.13.7 Creating a Target MCF (.met) for a Dynamic/Static Granule

A Target MCF file for a corresponding data granule can be created based on the information provided in the Source MCF file and the involved science software package (PGE).

In standalone or isolation testing of a PGE, the inputs needed by it will not have been Inserted by a previous PGE in the chain. This Insertion must be done manually. A Target MCF file for a corresponding data granule is required to run a standalone PGE. This way a dynamic data granule can be Inserted to the Science Data Server as if a PGE had produced it.

26.13.7.1 Creating a Metadata ODL File for a Static Granule

---

1 At the UNIX prompt on the AIT Sun, type cd WorkingPathname, then press the Enter key.
- Example: **cd /usr/ecs/{MODE}/CUSTOM/data/DPS/ODL/**
- The ***WorkingPathname*** *is* the full path name of the working directory containing the template metadata ODL file.

2 At the UNIX prompt on the AIT Sun, type **cp *StaticODLmet.tpl filename*.met**, then press the **Enter** key.
- The ***StaticODLmet.tpl*** is the file name of the template Target MCF.

- ↑ The ***filename*.met** is the file name of the Target MCF for this static file. The file name extension must be .met.
- ↑ This command will copy the template Target MCF to ***filename*.met**. For example, type **cp StaticODLmet.tpl CER11T.mcf.met**, then press the **Enter** key.

3 At a UNIX prompt on the AIT Sun, type **vi *filename*.met**, then press the **Enter** key.

- This command invokes the *vi* editor and reads in the Target MCF created above.

4 Edit the Target MCF with the specific information for the static data granule to be Inserted. The following guidelines should be followed when editing on the template MCF:

- The value for the ShortName object should be filled out with proper instrument name.
- The value for the Version ID object should be filled out with the proper version number.
- In the **INFORMATIONCONTENTCONTAINER** object enter the following:
  - ↑ The value for the **PARAMETERNAME** object of the class "1" should be filled out with the name of static data file.
  - ↑ The value for the **PARAMETERVALUE** object of the class "2" should be filled out based on the following guideline:
    - ↑ If the data granule is a coefficient file, a "C" followed by a numerical number n (n=1,2,….) will be used. Here n stands for the number of the coefficient file.
    - ↑ If the data granule is a MCF file, a "M" followed by a numerical number n (n=1,2,….) will be used. Here n stands for the number of the MCF file.

5 Save the changes made to the Target MCF (***filename*.met**) and exit the editor.

- ↑ The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, then press the **Enter** key.

26.13.8 Inserting Static Data Granules into the Data Server

Inserting a Static Data File:

- The following Servers/Services must be up and operational:
- **Science Data Server, Storage Management.**
- The following must have occurred between those Servers/Services:
- The ESDT of the static file must have been installed at the Data Server.

What the user must do before trying SSIT functionality:

- ↑ Create a metadata file for the static file to insert. To do this, an MCF (See "Getting an MCF in this section") must be gotten from the Data Server for

the ESDT of the file to insert. Mandatory fields are filled into the MCF, creating a metadata file.

- If the tool is NOT run from the SSIT Manager then go to the executables directory **(cd /usr/ecs/<MODE>/CUSTOM/utilities)**
- Source the buildrc file for the mode in which you are working (source .buildrc). Note that this only has to be done once per login.
- If the tool is NOT run from the SSIT Manager then go to the executables directory (**cd /usr/ecs/<MODE>/CUSTOM/bin/DPS)**

From the **SSIT Manager** choose **Tools** menu and then **Data Server** submenu. Choose **Insert Static File.**

The tool can also be executed by being in the **/usr/ecs/<MODE>/CUSTOM/bin/DPS and executing EcDpAtInsertStatic**

Shell script prompts user for information.

1 Enter in the location of the DpAtInsertStaticFile configuration file **(../../cfg/EcDpAtInsertStaticFile.CFG).**

2 Enter the MODE of operation (<*MODE*>). At the program prompt **mode (default ops)?**, or press **Enter** to take default.

3 Enter the short name of the ESDT (for the static file). This value is in the pdps database under the PlDataTypeMaster table and must be in the PGE ODL file.

- At the program prompt **ESDT name?** type ***ESDTShortName***, then press the **Enter** key. For example type: **MOD02LUT.**

4 Enter the version of the ESDT for the static file. This value is also in the pdps database under the PlDataTypeMaster table and must be in the PGE ODL file.

- At the program prompt **PGE version (default 1)?**, type ***PGEVersion***, then press the **Enter** key.
- The ***PGEVersion*** must match exactly the PGE version entered into the PDPS for this PGE.

5 Enter the science group for this static (this will be from the ODL created during Populating the PGE information in the Database).

- At the program prompt **Science group for Static file(one of{C,L,D,O} followed by a 3 digit number)?**, type ***ScienceGroupID***, then press the **Enter** key.
- ↑ The ***ScienceGroupID*** is an identifier used to define the file type as a coefficient file, a lookup table file, or a MCF. It distinguishes static granules of different types which share the same ESDT. For instance, for a coefficient file, use **C*n***, where number *n* could be 0, 1, 2…; this number *n* needs to be matched with the number *n* in the PGE_PGEName#Version.odl file. For an MCF. For example, type **C001**, press **Return.**
- ↑ The Science Group ID must match what was edited into the PGE metadata ODL file for that PCF entry.

6 At the program prompt **Is there more than one data file for this Static (Y = Yes, N = No)? (enter for default: N).** If there is only one data file, press **Return** and go to next step. If there are more than one data files, type **Y,** press **Return** and go to step 10.

7 At the program prompt **Single Static Filename to Insert (including FULL path)?**, type ***pathname*/*GranuleFileName***, press **Return**

- ↑ The ***pathname/GranuleFileName*** is the full path name and file name of the static data granule to be Inserted. For example, type **/home/MODIS/PGE10/MOD_PR28/coeff/emissivity.dat**, press **Return**.

8 At the program prompt **Associated ASCII Metadata Filename to Insert (including FULL path).** Type ***pathname/GranuleFileName.met*)?**, press **Return**.

- ↑ The ***pathname/GranuleFileName.met*** is the full path name and file name of the .met file for the associated static data granule to be Inserted. For example, type **/home/MODIS/PGE10/MOD_PR28/MOD28LUT.met** press **Return**.

9 At the program prompt **Directory where all data files and .met file exist (FULL path)?** Type ***pathname*** press **Return**.

- where ***pathname*** is the full path of the directory where all data files and .met file exist.
- Note for a multifile granule, the data files and .met file should be placed in the same working directory.

10 At the program prompt **Name of MFG file (enter to end list)?** Type in the ***GranuleFileName,*** one at a time and press **Return.** To end the list press **Return.**

- Where ***GranuleFileName*** is the names of the multifile granules.

11 At the program prompt **Associated ASCII Metadata Filename to Insert?** Type ***GranuleFileName.met***, press **Return.**

- Where ***GranuleFileName.met*** is the name of one **.met file** that is used with all data granules in the even of a multifile granule.
- The dynamic data granule will be Inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

12 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 2 through 9.

26.13.9 Inserting Dynamic Data Granules to the Science Data Server

In order for dynamic data files to be used both during the SSI&T and in production, this file must exist in the Data Server and be accessible by the local machine. A program called the Insert Test Dynamic File can be used for Inserting a dynamic data granule into the Data Server.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The ESDT's have been installed on the Data Server.
2. The Target MCF for this data granule has been created for the Insert.

To Insert a dynamic granule to the Data Server, execute the following steps:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert Test Dynamic**.

- An xterm with title "SSIT: PGE Test Dynamic Input File Insertion" will be displayed.

2 At the program prompt **Configuration filename? (enter for default: ../..cfg/EcDpAtInsertTestFile.CFG)**, press **Return**.

3 At the program prompt **ECS Mode of operations?**

- Type in the **<mode>** you are working in. For example, **TS1** or **OPS.** Press **Return.**

4 At the program prompt **ESDT short name for the file(s) to insert?** type ***ESDTShortName***, press **Return**

- The ***ESDTShortName*** is the ShortName of the ESDT descriptor file corresponding to this granule to be Inserted. For example, type **MOD021KM** press **Return**.

5 At the program prompt **ESDT Version for the file(s) to insert?** Type in the ESDT version and press **Return.**

6 At the program prompt **Is there more than one data file to this Dynamic Granule (Y = Yes, N = No)? (enter for default: N)?** If there are no multifiles for this ESDT, press **Return** and go to step 7. If there are more than one file for this granule go to step 9.

7 At the program prompt **Single Filename to Insert? (including FULL path)** type ***pathname/GranuleFilename***, press

- The ***pathname/GranuleFileName*** is the full path name and file name of the data granule to be Inserted. For example, type **/home/MODIS/PGE10/MOD021KM.A1996217.0014.002.hdf**, press **Return**.

8 At the program prompt **Associated ASCII Metadata Filename to Insert (including FULL path)** , Type ***pathname/GranuleFileName.met*** and press **Return**.

- ***pathname*** is full name of the path and ***GranuleFileName.met*** is the name of the associated .met file. For example, **/home/MODIS/PGE10/MOD021KM.met**
- The dynamic data granule will be Inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

9 At the program prompt **Directory where all data files and .met file exist (FULL path)?** Type ***pathname*** press **Return**.

where ***pathname*** is the full path of the directory where all data files and .met file exist.
Note for a multifile granule, the data files and .met file should be placed in the same working directory.

10 At the program prompt **Name of MFG file (enter to end list)?** Type in the ***GranuleFileName,*** one at a time and press **Return.** To end the list press **Return.**

where ***GranuleFileName*** is the names of the multifile granules.

11 At the program prompt **Associated ASCII Metadata Filename to Insert?** Type ***GranuleFileName.met***, press **Return.**

where ***GranuleFileName.met*** is the name of one .met file that is used with all data granules in the even of a multifile granule.

- The dynamic data granule will be Inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

12 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 2 through 8.

Example of a successful insertion of a Dynamic Input Data Granule into the Data Servers:

PGE Test Dynamic Input File Insertion **
Configuration filename? (enter for default:
../../cfg/EcDpAtInsertTestFile.CFG)
ECS Mode of operations? (enter for default: OPS)
OPS
ESDT name
MOD02H
ESDT Version (enter for default: 1)
0
Staged Filename to Insert? (including FULL path)
/home/emcleod/MCF/MOD02HKM.A1997217.1730.002.hdf
Associated ASCII Metadata Filename to Insert? (including FULL path)
/home/emcleod/MCF/MOD02H.met
Warning: Could not open message catalog "oodce.cat"
EcDpAtInsertTestFile: Process Framework: ConfigFile
../../cfg/EcDpAtInsertTestFile.CFG ecs_mode OPS
incomplete group entries in the configfile,using default G1
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default
Insert to Data Server successful:
ESDT Version = '0'
staged file = '/home/emcleod/MCF/MOD02HKM.A1997217.1730.002.hdf'
metadata file = '/home/emcleod/MCF/MOD02H.met'
Inserted at UR:
'UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:16:SC:M
OD02H:1757'
Hit return to run again, 'q <return>' to quit:

26.13.10 Science Server Archive Package (SSAP)

The SSAP is used to provide a record of the science software, documentation, and other related files stored at the DAAC. The SSIT SSAP GUI provides a method for grouping required data about a PGE.

The SSAP is not to be confused with the Delivered Algorithm Package (DAP) received from the SCF. Much of what is in the DAP will make it into the SSAP. The key difference is that SSAP

data is prepared after initial testing of the science software and will include data that reflects site integration as well as fixes required for performance at the DAAC.

The SSAP is made up of 2 different data types. The first data type is the Algorithm Package which contains metadata (name of the PGE, name of the instrument, date accepted, etc...) about the SSAP. The second data type is the source code, documentation, and test data which will be stored as a SSAP, with its own metadata in addition to the files. SSAP components such a source code will be tared to retain the directory structure.

The executables and static files are stored separately from the SSAP and will have their own data types (ESDTs).

The following is a list of tools, and or assumptions:

1. The SSIT Manager is running.
2. The PGE has been successfully built with the SCF and DAAC version of the Toolkit.

26.13.10.1 Creating an SSAP

The following Servers/Services must be up and operational:

Science Data Server, Storage Management.

The following must have occurred between those Servers/Services:

NONE.

What the user must do before trying SSIT functionality:

1 From the SSIT Manager choose **Tools** menu and then **Data Server** submenu. Choose **SSAP Editor.**
The GUI starts. Note that it will first query Data Server for a list of SSAPs that have previously been created. This list will appear in the window at the center (if any SSAPs already exist). Current SSAP field will be blank, and only Refresh and Create buttons will be active. All three tabs (Main, Files, and Metadata) will be active.
The SSAP GUI will be displayed.

2 Click on the **Create** to create a new SSAP.

- The **Create SSAP** window appears. If no OK button is visible, resize the window such that the OK button is visible.

3 Enter the name of the SSAP in the first field . Enter SSAP version in the second field. Note that version has a limit of 20 characters.

4 Click OK and the window disappears

5 On the main GUI, the SSAP created (what was entered in the step above) will appear. Current SSAP is now set to that value. All buttons are now active.

5 To set up the SSAP components, click on the **File List** tab.

- The File List Tab displays files in the local directory to the left and files in the selected SSAP component to the right. On the bottom left is a directory listing and a

method to move through the directory tree on the local machine. Delete and Reset buttons—both active – are to the right.

6 To select a file in the left column, click on the **File Type** button, highlight a file (or files) and click on the **Add** arrow button to add the files..
The files selected to be added will be displayed in the right column.
T o change directories (and thus add files from other directories to the SSAP component), click on the listing in the window on the bottom left of the GUI. The ".." is to go up one directory level. A single click with move to the directory chosen and change the display to show the directories under the new current directory. Note that the list of files in the upper left window changes to show the files within the current directory.

7 To add metadata for the new SSAP, select the **Metadata** tab.
The Metadata window will be displayed.
The Metadata Tab displays the metadata for the new SSAP. Only the Name and Version will be filled in automatically. The rest of the fields will have default information. While the SSAP can be submitted with the default information, it is wise to fill in valid values. To change a value:

8 To change the default information, click on the **Edit Assoc Collections** button.
The **Edit Associated Collections** window displays a list of associated collections and fields for the entry of new ShortNames and Versions.

9 Enter a ShortName, and version (of the ESDT that has been installed in the Data Server) - must be eight or fewer characters. Note that the Data Server will verify if the Shortname exists.

10 Enter the version (of the installed ESDT).
Then select the **OK** button. Select **Done** to close the window.

11 To save the updated metadata, click **Save** on the **Metadata** tab**.**

12 To get back to the **Main** tab**,** select the **Main** tab button.

13 To submit the new SSAP to the Data Server, select the **Submit** button.
When the SSAP has been submitted, the **SSAP Successfully inserted to the Data Server** prompt will appear.

26.12.10.2 Updating an SSAP

The following Servers/Services must be up and operational:

Data Server, Storage Management.

The following must have occurred between those Servers/Services:

An SSAP must have already been inserted to the Data Server.

What the user must do before trying SSIT functionality:

The SSAP Editor has been used to insert an SSAP to the Data Server.

What must be done via SSIT tools:

If SSAP Editor is not running, use the directions from the first 2 paragraphs of (Creating an SSAP) to bring up the SSAP GUI. Note that the added SSAP should appear in the window of the Main tab.

If the SSAP Editor is already running, the added SSAP should appear in the window of the Main tab.

---

1 Click on added SSAP in the main display.

2 Click on the Metadata tab to update the SSAP.
The Metadata Tab displays the metadata for the SSAP. All fields will be set to the values entered when the SSAP was created, and the Algorithm Name field will be grayed out (because it may not be updated). If you want to create a new SSAP from the an existing one, go back to the Main tab and hit the Create With button.

3 Click on the Algorithm Version field (currently called Algorithm Description) and enter a new version (different from what is in the field when the tab is clicked).

4 Update any other fields that you wish to change. You can even add a new Associated Collection by clicking on the Assoc Collection button and following the steps described in Creating an SSAP.

5 Before you leave the Metadata tab, click Save to save the updated metadata.

6 Click on the File List tab to set up new SSAP components.
The File List Tab displays files in the local directory to the left and files in the selected SSAP component to the right. On the bottom left is a directory listing and a method to move through the directory tree on the local machine. Delete and Reset buttons—both active –are to the right.

7 Click on the File Type button to select the additional SSAP component to manipulate.
Choose one of the menu items.
Select a file (or files) from the left window to add to the component.
Click the Add Arrow button to add the files. They will appear in the right window because they are now part of that SSAP Component.
Click Main to get back to the Main tab.
On the Main tab:

Click Submit to send the new SSAP to Data Server. When finished, a message should pop up that says "SSAP Successfully inserted to the Data Server".

26.13.11 PGE Checkout

The following Servers/Services must be up and operational:

NONE.

The following must have occurred between those Servers/Services:

NONE.

What the user must do before trying SSIT functionality:

In normal SSIT (at the DAACs) the DAP would be untared, and the source code recompiled and tested.

What must be done via SSIT tools:

Since SSIT is just a calibration of various tools, there is no specific order for which they must be run. All tools can be started from the SSIT Manager and can be executed on their own.

For SparcWorks (for code analysis and debugging), choose Tools menu and then Code Analysis submenu. See SparcWorks manuals for SparcWorks operation.

For various office tools, choose Tools menu and the Office Automation submenu. Choose from MS Windows (a simulator to allow the user to run Windows programs), Ghostview (a viewer), Netscape (for web access), Acrobat (for document viewing), and DDTS (for problem reporting).

For Standards checkers, choose Tools and then the Standards Checkers submenu.

FORCHECK is a COTS Fortran language checking program.

The Prohibited Function Checker will examine source code for functions that are not permitted On Prohibited Function Gui, choose Analyze to select files to examine. Hit the Ok button once selections are made and a message at the top of the Gui will indicate if prohibited functions have been found. If prohibited functions HAVE been found, use the View button to view the source code with the prohibited call. The Help button gives further information on how to work the Gui.

The Process Control File Checker examines selected PCFs and highlights any errors. The Process Control File Gui allows the user to work through the directory structure on the local machine and select PCFs to be checked. Click the Check PCF button to check a selected PCF. Again, the Help button provides more information.

26.13.12 Placing the Science Software Executable (SSEP) on the Data Server

In order to be able to run a PGE within the ECS system, the EXE TAR file has to be inserted to the Science Data Server. This tar file consists of all files needed to run a PGE, except for input data files. This includes the executables, any scripts, and the SDP Toolkit message files.

26.13.12.1 Assembling a Science Software Executable Package (SSEP)

This section describes how to assemble a Science Software executables Package (SSEP) and create a corresponding Target MCF. A SSEP is a UNIX tar file which contains PGE executables and SDP Toolkit message files.

In order to Insert a PGEEXE tar file into the Science Data Server, a corresponding Target MCF (.met) must be generated before insertion. Such an ASCII metadata ODL file can be obtained by editing an existing template ODL file with the information of the specific PGE. The following procedures describe how to assemble a PGEEXE tar file and create an ASCII metadata ODL file.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. PGE executables and message files required by this PGE are available to make a SSEP.

To create an SSEP, execute the steps that follow:

1 At the UNIX prompt on an AIT Sun, type **mkdir *SSEPpathname***, press **Return**.

- The ***SSEPpathname*** is the full path name of a *new* directory which will contain all the files to be placed into the SSEP as well as the SSEP itself.
- It is recommended that ***SSEPpathame*** be named with a convention that indicates the PGE for which a SSEP will be created. For example, type **mkdir PGE35.ssep**, press **Return**.

2 At the UNIX prompt on the AIT Sun, type **cd *SSEPpathname***, press **Return**.

- The ***SSEPpathname*** is the directory name of the new directory created in step 1.

3 At the UNIX prompt on the AIT Sun, type **cp *pathname/file1 pathname/file2 … pathname/filen* .**, press **Return** (note the "dot" and then space at the end of the command).

- The ***pathname/file1, pathname/file2,…pathname/filen*** represents a list of path names and file names (delimited by spaces) to copy into the current directory, ***SSEPpathame*** (the "dot" represents the current directory and must be last in the command).
- For example, type **cp /data/MODIS/pge/PGE35.exe /data/MODIS/mcf/mod35.mcf /data/MODIS/MOD_13453 .**, press **Return** (note the space and then "dot" at the end of the command).
- The files copied into this directory should be the PGE executable, any shell scripts or other executables that are part of the PGE and SDP Toolkit message files.
- Files can be individually copied into the ***SSEPpathame*** directory. For example, type **cp /data/MODIS/pge/PGE35.exe .**, press **Return** (note the space and then "dot" at the end of the command). Repeat for each file needed in the SSEP for this PGE.

4 At the UNIX prompt on the AIT Sun, type **tar cvf *SSEPfilename*.tar ***, press **Return**.

- The ***SSEPfilename*.tar** is the file name for the SSEP tar file. The file name extension .tar is recommended but not required.

- The asterisk (*****) is a file name wildcard that represents all files in the current directory. This will place all files in the SSEP tar file.
- Once created, the contents of the SSEP tar file can be viewed by typing **tar tvf *SSEPfilename*.tar**, press **Return**.
- Do not apply compression (*e.g.* UNIX compress or gzip) to the tar file.

5 At the UNIX prompt on the AIT Sun, type **cp *filename.met.tpl filename.met,*** press **Return**.

- The ***filename.met.tpl*** is the file name of the template Target MCF for this SSEP. If a template is not available, see Appendix D or use one used for another SSEP.
- The ***filename.met*** is the file name of the Target MCF to be tailored for this SSEP.

6 At the UNIX prompt on the AIT Sun, type **vi *filename.met***, press **Return**.

- The ***filename.met*** is the Target MCF for this SSEP.
- This command invokes the *vi* editor. Edit the ***filename*.met** with the specific information for the SSEP to be inserted.
- The following guidelines should be followed when editing on the Target MCF (***filename*.met**):
  - The value for the VERSIONID object should be filled out with the proper PGE version. For example: "1" .
  - In the INFORMATIONCONTENTCONTAINER object,
    - The value for the PARAMETERNAME object of the class "1" should be filled out with the PGE name. For example: "BTS".
    - The value for the PARAMETERNAME object of the class "2" should be filled out with the PGE Science Software Version. For example: "1".
    - The value for the PARAMETERNAME object of the class "3" should be filled out with the Platform Name. For example: "IRIX".
    - The value for the PARAMETERNAME object of the class "4" should be filled out with the Platform Version. For example: "6.2".
    - The value for the PARAMETERNAME object of the class "5" should be filled out with the date to perform the Insertion. For example: "970319".
    - The value for the PARAMETERNAME object of the class "6" should be filled out with the time to perform the Insertion. For example: "14:45:00".

7 Save the changes made to the SSEP's Target MCF (***filename*.met**) and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.

For other editors, refer to that editor's documentation

Procedures continued

1 At the UNIX prompt on the AIT Sun, type **mkdir *SSEPpathname*** then press the **Enter** key. For example, type **mkdir MOD35.ssep, press Enter.**

The ***SSEPpathname*** is the full path name of a *new* directory which will contain all the files to be placed into the SSEP as well as the SSEP itself.

2 At the UNIX prompt on the AIT Sun, type **cd *SSEPpathname***, then press the **Enter** key.

The ***SSEPpathname*** is the directory name of the new directory created in step 1.

3 At the UNIX prompt on the AIT Sun, type **cp *pathname/file1 pathname/file2 … pathname/filen* .**, then press **Enter** (note the space then the "dot" at the end of the command).

The ***pathname*** is the location of the files. The ***file1, file2,…filen*** represents a list of file names (delimited by spaces) to copy into the current directory, ***SSEPpathame*** (the "dot" represents the current directory and must be last in the command). For example, type **cd /data/MODIS/pge/MOD35.pge /data/MODIS/mcf/MOD35.mcf /data/MODIS/MOD_13453 ., press Enter.** (note the space then the "dot" at the end of the command).

For the synthetic PGE, only the executable needs to be copied.

4 At the UNIX prompt on the AIT Sun, type **tar cvf *SSEPfilename*.tar ***, then press the **Enter** key.

The ***SSEPfilename*.tar** is the file name for the SSEP tar file.

The file name extension .tar is recommended but not required.

The asterisk (*****) is a file name wildcard that represents all files in the current directory which will place all files in the SSEP tar file.

Once created, the contents of the SSEP tar file can be viewed by typing **tar tvf *SSEPfilename*.tar**, then press the **Enter** key.

5 At the UNIX prompt on the AIT Sun, type **cp *filename.met.tpl filename.met,*** then press the **Enter** key.

The ***filename.met.tpl*** is the file name of the Target MCF for this SSEP.

For the synthetic PGE, the **met** file has already been renamed and modified for use by the student when the file was unpacked.

6 At the UNIX prompt on the AIT Sun, type **vi *filename.met***, then press the **Enter** key.

The ***filename.met.tpl*** is the Target MCF for this SSEP.

7 Edit the ***filename*.met** with the specific information for the SSEP to be inserted.

The value for the **VERSIONID** object should be filled out with the proper PGE version.

In the **INFORMATIONCONTENTCONTAINER** object enter the following:

The value for the **PARAMETERNAME** object of the **class "1"** should be filled out with the PGE name. The synthetic PGE should be "userid".

The value for the **PARAMETERNAME** object of the **class "2"** should be filled out with the PGE Science Software Version.

The value for the **PARAMETERNAME** object of the **class "3"** should be filled out with the Platform Name.

The value for the **PARAMETERNAME** object of the **class "4"** should be filled out with the Platform Version.

The value for the **PARAMETERNAME** object of the **class "5"** should be filled out with the date to perform the Insertion.

The value for the **PARAMETERNAME** object of the **class "6"** should be filled out with the time to perform the Insertion.

8 Save the changes made to the SSEP's Target MCF (***filename*.met**) and exit the editor.

The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, then press the **Enter** key.

26.13.12.2 Inserting a Science Software Executable Package onto the Data Server

. Science software, like any other data that are managed in the ECS, must be placed on the Science Data Server. A program called the Insert EXE TAR Tool can be used for Inserting a Science Software Executable Package into the Data Server.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The ESDT called PGEEXE has been installed on the Science Data Server.
2. A Target MCF (.met) for this PGEEXE tar file has been created for the Insert.

1. The PGEEXE tar file has been created .\
2. The following Servers/Services must be up an operational:

Science Data Server, Storage Management.

To Insert the SSEP to the Science Data Server, execute the steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert EXE TAR**.

- An xterm with title "SSIT: PGE Executable Tar File Insertion" will be displayed.

2 At the program prompt **Configuration filename? (enter for default: ../../EcDpAtInsertExeTarFile.CFG)**, press **Return**.

3 At the program prompt **ECS mode of operations?**, Type **<*mode*>** press **Return.**

- **<*mode*>** can either be **OPS** or **TS1**.

4 At the program prompt **Name of PGE?**, type ***PGEName***, press **Return**.

- The ***PGEName*** is the name of the PGE for which this static granule is being Inserted. For example, type **PGE01**, press **Return**.
- The ***PGEName*** must match exactly the PGE name entered into the PDPS for this PGE.

5 At the program prompt **Science software version of PGE?**, type ***SSWversion***, press **Return**.

- The ***SSWversion*** is the version of the science software which is being Inserted in this SSEP. Press **Return** to accept the default or enter in a version and press **Return**.

6 At the program prompt **Staged filename to insert (including Full path)?**, type ***pathname*/*SSEPFileName***, press **Return**

- The ***pathname/SSEPFileName*** is the full path name and file name of the SSEP tar file to be Inserted. For example, type **/data/MOD35/ssep/PGE35_1.tar**, press **Return**.
- The SSEP tar file must not be compressed (*e.g.* with UNIX compress or gzip).

7 At the program prompt **Associated ASCII metadata filename to insert (including Full Path)? *pathname/SSEPFileName.met*)?**, press **Return**.

- The default is the file name of the granule to insert with the .met file name extension. If the default is not correct, then the file name of this file must be entered.

8 At the program prompt **Top level shell filename within tar file?**, type ***ExecFileName***, press **Return**.

- The ***ExecFileName*** is the file name of the top level executable or script within the SSEP tar file. It should be the same as was entered into the PDPS/SSIT Database Update GUI.
- The SSEP will be Inserted to the Science Data Server.

9 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 3 through 8.

Example of a successful insertion of a SSEP EXE TAR:

PGE Executable Tar File Insertion Script

Configuration filename? (enter for default:../../cfg/EcDpAtInsertExeTarFile.CFG)

ECS Mode of operations? (enter for default: OPS)

OPS

Name of PGE? (enter for default: PGE07)

PGE07

Science software version of PGE? (enter for default: 0)

0

Staged filename to insert (including FULL path)? (enter for default:

/home/emcleod/SSEP/MODPGE07.tar)

Associated ASCII metadata filename to insert (including FULL path)? (enter for

default: /**home/emcleod/SSEP/MOD_PR10.tar.met**)

Top level shell filename within tar file? (enter for default:

/home/emcleod/SSEP/MOD_PR10.exe)

MOD_PR10.exe (note: this entry is done a second time) Note: If you get **core dump**, execute using "dbx command: type in: **dbx filename .exe**. This will help isolate error message that caused core dump.

Warning: Could not open message catalog "oodce.cat"

```
EcDpAtInsertExeTarFile: Process Framework: ConfigFile
../../cfg/EcDpAtInsertExeTarFile.CFG  ecs_mode OPS
Performing INSERT.....
incomplete group entries in the configfile,using default G1
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default
Insert to Data Server and PDPS database update successful for:
  PGE name = 'PGE07'
  Ssw version = '0'
  ESDT = 'PGEEXE'
  ESDT Version = '0'
  staged file = '/home/emcleod/SSEP/MODPGE07.tar'
  metadata file = '/home/emcleod/SSEP/MOD_PR10.tar.met'
  Top level shell name = 'MOD_PR10.exe'
Inserted at UR:
'UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:14:LM:PGEEXE:1787'
Hit return to run again, 'q <return>' to quit:
```

26.14 PGE Planning Processing and Product Retrieval

26.14.1 Using the Production Request Editor

When standalone tests (Run from the command line) have completed successfully and information about the PGE has been entered into the PDPS Database (through PGE registration), the PGE is ready to be run through the automated ECS PDPS environment.

To process Science data, a Production Request (PR) must be submitted to the ECS system. The Production Request Editor GUI accomplishes this function. Only one PR may be submitted at a time. A single PR is exploded by the PDPS into one or more jobs called Data Processing Requests (DPRs). The number of DPRs that are created for a single PR is determined by the number needed to cover the requested time interval, orbital extent and tile schema. Some PRs may only require one DPR.

26.14.2 Invoking the Production Request

Currently, the Production Request Editor is invoked from a command line script. In the future, this will be done by clicking on the icon for the PR Editor on the ECS Desktop. Once the Production Request Editor is invoked, it brings up a screen with five tabs at the top for selection

(as shown in Figure 26.14.3-1). The first tab is labeled “Planning”. Selection of this tab displays a list of four capabilities available for the PR Editor by selecting the other tabs at the top of the primary GUI screen: PR Edit, PR List, DPR View, and DPR List.

********* Please be advised: Only one user per mode can be executing a DPR at a time. If more than one occurs the system will cancel the remaining DPR’s. Problem discovered when chained PGE’s failed to kickoff. *********

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The PGE has been registered in the PDPS Database.
2. The PGE has been successfully compiled and linked with the DAAC version of the SDP Toolkit.
3. The required servers are up and running.

To invoke the Production Request Editor GUI, execute the procedure steps that follow:

1 In any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the PLS host.

- It is recommended that this procedure begin within a new command shell on a PLS Host.

2 Set the DISPLAY environment variable. At the UNIX prompt on the PLS host (e.g. **p0pls01**), type **setenv DISPLAY *terminal_id***, press **Return**.

3 Set the DCE environment variable. At the UNIX prompt on the PLS host (e.g. odyssey), type **dce_login *DCE_user_name DCE_password***, press **Return**.

4 Set the UNIX environment variable. At the UNIX prompt on the PLS host, change to the directory where the scripts are located (e.g. **cd** /usr/ecs/TS1/CUSTOM/utilities), then

- Type **setenv MODE *mode*** (e.g. TS1).
- Type **source *environment_setup_file*** (e.g. EcCoEnvCsh for C shell users).

5 At the UNIX prompt on the PLS host (e.g. odyssey), under the directory where the scripts are located (e.g. **cd** /usr/ecs/TS1/CUSTOM/utilities), type **EcPlPRE_IFStart *mode application_id* &**, then press **Return**.

- The ***mode*** is the operations mode (e.g. TS1).
- The ***application_id*** is a numerical number (e.g. 1).
- For example, type **EcPlPRE_IFStart *TS1 1* &**, press **Return**.
- Various messages from the Production Request Editor may appear in this window as it is running. For this reason, avoid using this window for other tasks until the Production Request Editor has terminated.

6 In the Production Request Editor, click on one of the tabs PR Edit, PR List, DPR View, or DPR List corresponding to desired task.

- To define a new Production Request or edit a Production Request, click on PR Edit. Proceed to Section Defining a New Production Request.
- To review or list a Production Request, click on PR List.
- To view or inspect a Data Processing Request, click on View.
- To review or inspect a Data Processing Request, click on List.

7 When tasks are completed in the Production Request Editor GUI, click on the **File** menu, then choose **Exit**.

- The Production Request Editor will disappear.
- Refer to Section (Troubleshooting and General Investigation) if fail to bring up the Production Request Editor GUI.

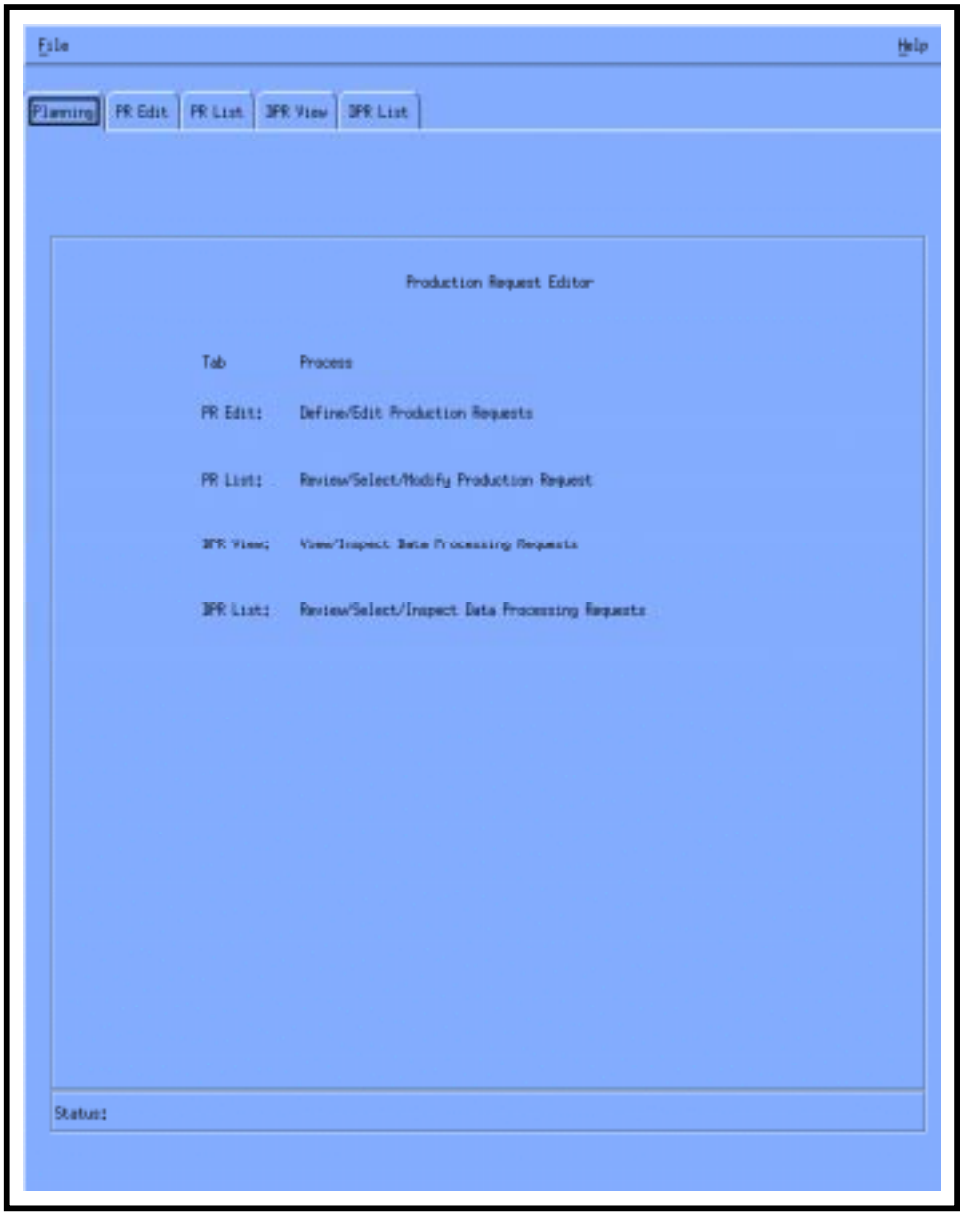


Figure 26.14.3-1. Production Request Editor Introductory GUI

26.14.3 Defining a New Production Request

A Production Request (PR) is a request for data production of granules between a start date/time and an end date/time. A PR will explode into one or more Data Processing Requests (DPR) depending upon the time interval involved. Each DPR corresponds to the execution of a PGE. Therefore, a PR results in the execution of a PGE one or more times. Only one PGE is involved in a single Production Request.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

- The user has selected the **PR Edit** tab from the Production Request Editor .
- The PGE involved in the Production Request has been registered in the PDPS database.

On workstation **x0pls##**, at the UNIX prompt in a terminal window, type as in step 1 below your user id and password.

NOTE: The **x** in the workstation name will be a letter designating your site: **g** = GSFC, **m** = SMC, **l** = LaRC, **e** = EDC, **n** = NSIDC, **o** = ORNL, **a** = ASF, **j** = JPL; the ## will be an identifying two-digit number (e.g.,**g0pls02** indicates a Planning Subsystem (PLS) workstation at GSFC).

Prior to the rlogin, enter **setenv DISPLAY <local_workstation IP address>:0.0.** The **<ipaddress>** is the ip address of **x0pls##,** and **xterm** is required when entering this command on a Sun terminal.

To define a new Production Request, execute the procedure steps that follow:

1 From the Production Request Editor GUI, click on the **PR Edit** tab.

- The PR Edit page will be displayed as shown in **Figure.** 26.14.3-2.

2 In the field labeled **PR Name:**, enter **New** or verify that **New** is already entered as the default.

3 The PGE for the Production Request must be selected from a list. To do this, click on the **PGE…** button.

- A GUI labeled **PGE Selection** will be displayed within which registered PGEs will be listed. The appropriate PGE can then be selected by clicking on it and then on the **OK** button.
- The selected PGE will then be used to populate **Satellite Name**, **Instrument Name**, **PGE Name**, and **PGE Version** fields of the main GUI.

4 In the field labeled **Priority:**, enter ***priority***.

- The ***priority*** is the priority to be assigned to this Production Request in the range 0 through 999 with 0 being the highest priority and 999 the lowest. For example, enter **40**.

5 In the Production Request Editor GUI, a **Duration** option is selected automatically based on the PGE registered into the PDPS. Two options are provided:

- **UTC Time** for time range.
- **Orbit** for orbit number range.

6 In the Production Request Editor GUI, enter ***StartDate*** and ***StartTime*** in fields labeled **Begin**, respectively.

- The ***StartDate*** and ***Starttime*** are the start date and time of the Production Request and should be entered in the mm/dd/yy and hh:mm:ss formats.

7 In the case of UTC Time duration, enter ***EndDate*** and ***EndTime*** in fields labeled **End**, respectively.

- The ***Enddate*** and ***EndTime*** are the end date and time of the Production Request and should be entered in the mm/dd/yy and hh:mm:ss formats.

8 In the case of Orbit duration, enter ***StartOrbit*** and ***EndOrbit*** in fields labeled **From** and **To**, respectively.

- The ***StartOrbit*** and ***EndOrbit*** are the orbit range of the Production Request.

9 Optionally, enter ***Comment*** in field labeled **Comment:**.

- This comment will be displayed whenever this Production Request is brought up and viewed.

10 When Production Request is complete, click on **File** menu and select **Save As….**

- A GUI labeled **File Selection** will be displayed.
- In the field labeled **Selection**, enter a user-defined name to be assigned to the Production Request. Then click on the **OK** button. A message box will be displayed stating "Production Request Explosion into DPRs ok, *n* DPRs Generated", where *n* will be the number of DPRs (e.g. a 2-hr PR time will generate 24 DPRs for the 5-min processing period). Click on the **Ok** button. A second message box stating "Write to Database of Production Request ok"; again click **Ok**.

Note that you will not be allowed to enter a PR name that already exists. PR names that already exist will be displayed in the main window. The Production Request will then be saved under the name specified.

- Refer to Section (Troubleshooting and General Investigation) if fail to generate the DPRs.

11 When tasks are completed with the Production Request Editor GUI, click on the **File** menu, then choose **Exit**.

- The Production Request Editor GUI will disappear.

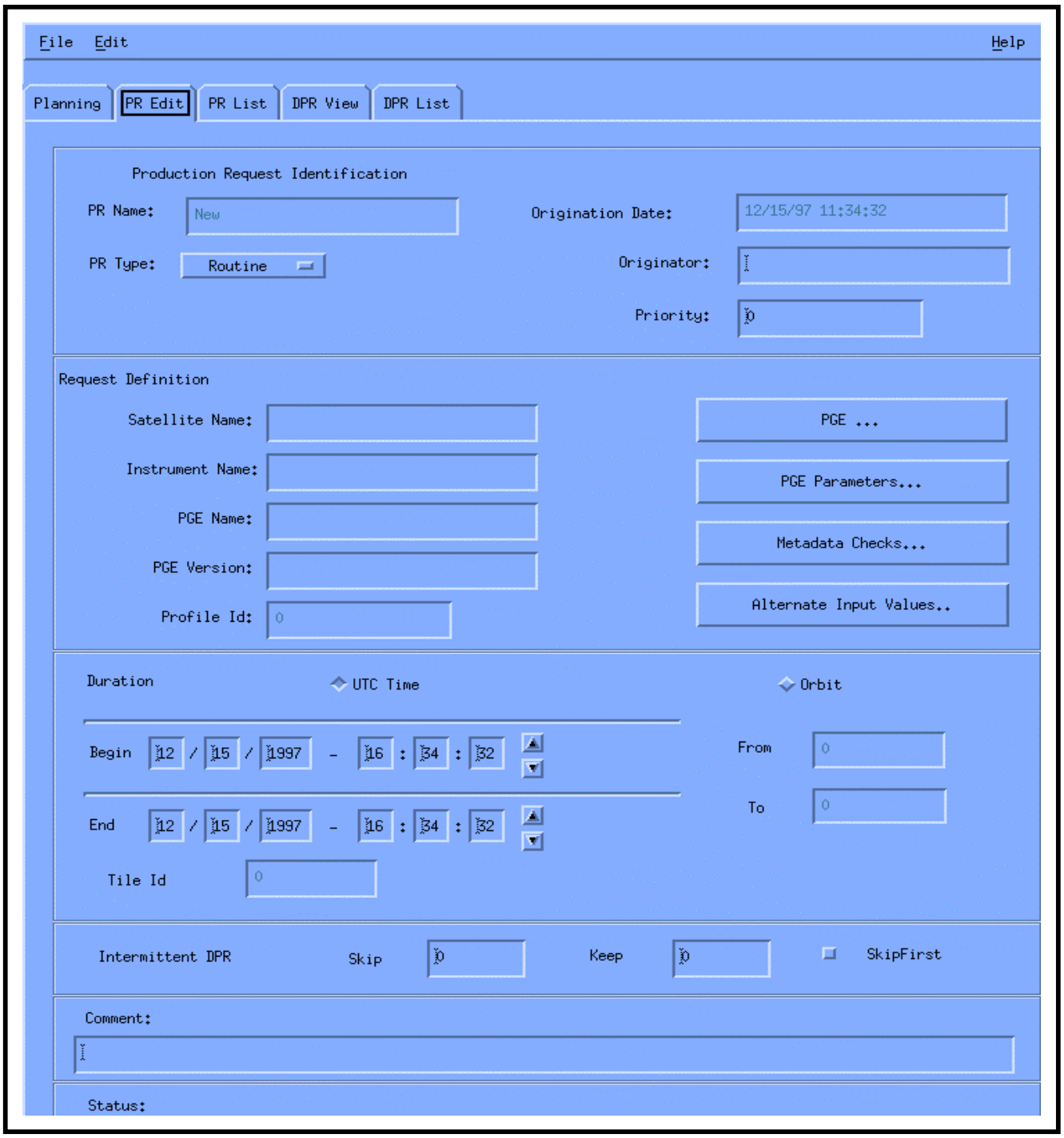


Figure 26.14.3-2. Production Request Editor GUI(Planning)

26.14.4 Processing

Once a candidate plan has been activated, each of the DPRs will result in subscriptions to the Data Server for the data needed. A request will go to the Data Server asking for notification when the required input data arrives.

Planning knows what data to request from the Data Server because the PDPS database stores this information as determined by the ESDT for each PGE. When the Data Server receives new data, it routinely checks to see if there are any outstanding subscriptions. If there are subscriptions, Planning will be notified. Once the input data required by a DPR becomes available, the DPR can be queued for processing.

Staging - The Data Processing Subsystem requests that the required input data, PGE (binary executables and shell scripts) and SDP Toolkit files be placed on a disk set aside for processing.

Process Control File (PCF) - establishes a linkage between logical Ids that the science software uses and the physical files that exist on the staging disk.

After the PGE has completed, the DPS will deallocate resources.

A Production History file will be created and will contain information concerning the conditions that the data products were generated by the PGE.

26.14.4.1 Viewing Production Requests

A Production Request (PR) is a request for data production of granules between a start date/time and an end date/time. A PR will explode into one or more Data Processing Requests (DPR) depending upon the time interval involved. Each DPR corresponds to the execution of a PGE. Therefore, a PR results in the execution of a PGE one or more times. Only one PGE is involved in a single Production Request.

This procedure describes how to view PRs that have already been defined . It assumes that the **PR List** tab has been selected from the Production Request Editor.

The information listed for each PR is:

- PR Name - The name assigned to the Production Request when it was defined.
- PGE ID - The name of the PGE involved in the PR.
- Priority - The priority (0 - 99) of the PR assigned when it was defined.
- Start - The start date and time of the PR.
- End - The end date and time of the PR.
- Comment - Any comment that was entered when the PR was defined.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The user has selected the **PR List** tab from the Production Request Editor.

1 From the Production Request Editor GUI, click on the **PR List** tab.

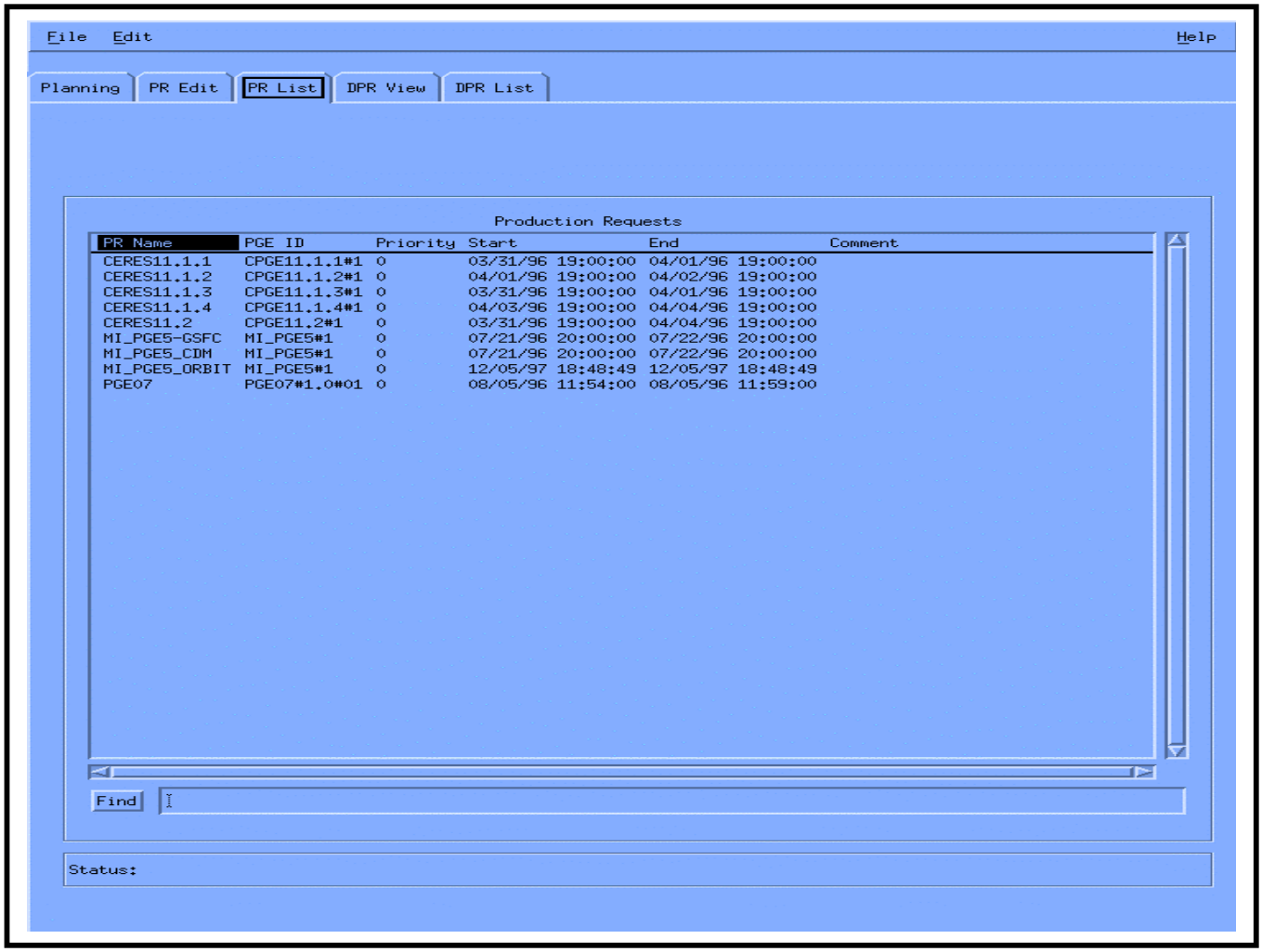
- The PR List page will be displayed as shown in **Figure.** 26.14.4-1.

2 View the listed PRs. Optionally, find a PR by entering a search string in the field next to the **Find** button and then clicking on the **Find** button.

3 To modify a PR listed, click on the PR in the list and from the **File** menu select **Save As….**

- In the **File Selection** GUI, replace the current PR name shown in the **Selection** field with a new PR name. Then click on the **OK** button.
- When modifying an existing PR, it must be saved under a new PR name.
- Next, click on the **PR Edit** tab. The PR Edit page will be displayed with fields populated from the existing PR name, but having the new PR name chosen above.
- See Section on using the PR Edit page and saving any changes made

Figure 26.14.4-1. PR List GUI.



26.14.4.2 Viewing Data Processing Requests

Clicking on the DPR View GUI tab displays a list of all DPRs for all PRs entered into the system.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

1 From the PR Editor GUI, click on **DPR View** tab. The following information will be displayed:

- Data Processing Request Identification.
- PGE ID and its parameters.
- Request Data and Status.

26.14.4.3 Listing Data Processing Requests

Selection of one PR on the PR List by highlighting it and then clicking on the DPR List tab, brings up a detailed display of all DPRs associated with the selected PR. These may be examined in order to develop production plans and schedule jobs.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

1 From the PR Editor GUI, click on **DPR List** tab**.** The following information is displayed:

- DPR Id.
- PGE Id.
- PR Name.
- Data Start Time.
- Data Stop Time.

26.14.5 Using the Production Planning Workbench

The Production Planner uses the Production Planning Workbench to create new production plans and display a planning timeline.

26.14.5.1 Using the Planning Workbench to Run a PGE

Once a PGE has been fully registered, its test data files have been inserted to the Science Data Server, and a single Data Processing Request (DPR) has been generated, the Planning Workbench can be used to plan for one execution run of a single PGE.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The required UNIX environment variables have been set properly.
2. The required servers of the ECS System are up and running.

3. A DPR has been generated successfully.

1 **Telnet to (PDPS) odyssey** or from the **SSIT Manager**, click on the **Tools** menu, then choose **Xterm**. Then telnet to a PLN Host.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the PLN Host.
- It is recommended that this procedure begin within a new command shell on a PLN Host.

2 login: **ID**
password:

3 *Login to DCE* **(dce_login DCE_user_name DCE_password and then press Enter Key.***), **setenv DISPLAY ….:0.0***

6. Change the working directory to the location where the script for starting the Autosys is stored. (e.g**. cd /usr/ecs/<mode>/CUSTOM/utilities).**
7. Type**: source EcCoEnvCsh**

4 At the UNIX prompt on an PLN Host, Type **EcPlAllStart <mode> <application_id>**, press **Return**.

- The ***mode*** is one of modes used in the ECS system. e.g. TS1.
- The ***application_id*** is a numerical number. e.g. 1.
- For example: **EcPlAllStart TS1 1, Return.**
- A Planning Workbench GUI will be appeared as shown in **Figure 26.14.6-1.**

5 In the Planning Workbench GUI, go to the subwindow labeled **Unscheduled** and click on a Production Request name.

- The Production Request name is the name under which the PR was saved.
- The PR name entry will be highlighted.

6 In the Planning Workbench GUI, click on the button next to the label **Schedule** (the button has an inverted triangle on it).

- The PR highlighted in step 3 will appear in the subwindow labeled **Scheduled**.

7 In the Planning Workbench GUI, click on the **Activate** button

- A small GUI labeled **Plan Activation** will be displayed.

8 In the Plan Activation GUI, set the time in the time field forward to allow ample time for the PGE to run. Then click on the **Ok** button.

- All that is necessary is for there to be sufficient time for the PGE run. There is no penalty for allowing *too* much time.
- The Production Request thus planned will be submitted to processing and its progress can be monitored with AutoSys.

10 When tasks are completed with the Planning Workbench GUI, click on the **File** menu, then choose **Exit**.

- The Planning Workbench GUI will disappear.

Important Note **:** The creation of Production Requests and Plans requires close coordination among all who are using the same mode in SSIT. Otherwise, the submittal of a Plan may prevent a waiting DPR from starting. In particular, when submitting a new Plan in a mode being shared with others, one should:

- Ask everyone else if they have a DPR awaiting data as part of a chain.
- Check the PDPS database table PlDataProcessingRequest for any DPRs that are in state CQ_HOLD and include these in the new Production Request and Plan.
- Also, include any DPRs (except those marked SUCCESS) upon which the given DPR depends in the new Production Request and Plan.

26.14.6 Creating and Activating of a Production Plan

The Production Planner creates a plan for production data processing at the DAAC by selecting specific PRs whose DPRs are to be run. The planning tool provides a forecast of the start and completion times of the jobs based upon historical experience in running these PGEs. Through the planning tool, when the generated plan is "activated," the information included in the plan is transferred to the Data Processing subsystem and loaded into the Platinum AutoSys tool where production processing is managed.

The Production Planner creates the plan by selecting PRs from two lists of PRs, i.e., the list of available "Unscheduled" PRs and the list of "Scheduled" PRs. Using arrow buttons, the Production Planner moves the PRs between lists until the "Scheduled" list contains the desired set of PRs that define the new plan. Only one user can use the **Planning Work Bench** at a time. It is recommended for SSI&T that only one person do the planning for the group.

Before creating a new production plan the Production Planner must have available the following information:

Name of the plan.
Comments (if any).
PRs to be included in the new production plan.

1 Log into one of the pln sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

Prior to the remote login, enter **setenv DISPLAY <**local workstation IP address>**:0.0** where the local workstation IP address represents the IP address you where you are located**.**

You may need to setup the terminal so that the remote host is displayed on your screen (Sun machine). This is done by clicking on the **Application Manager** icon (the file drawer located at the bottom of the screen), followed by the **Desktop Tools** icon**,** followed by the **Terminal Console** icon

3 At a UNIX prompt type cd to the directory where the scripts are located. (e.g. **/usr/ecs/TS1/CUSTOM/utilities**).

4 At a UNIX prompt type **setenv DISPLAY *hostname*:0.0**

5 At a UNIX prompt on the PLN host (e.g. p0pls01), type **EcPlPRE_IFStart mode 3** a **Planning Workbench** GUI is displayed (Figure 26.14.6-1)

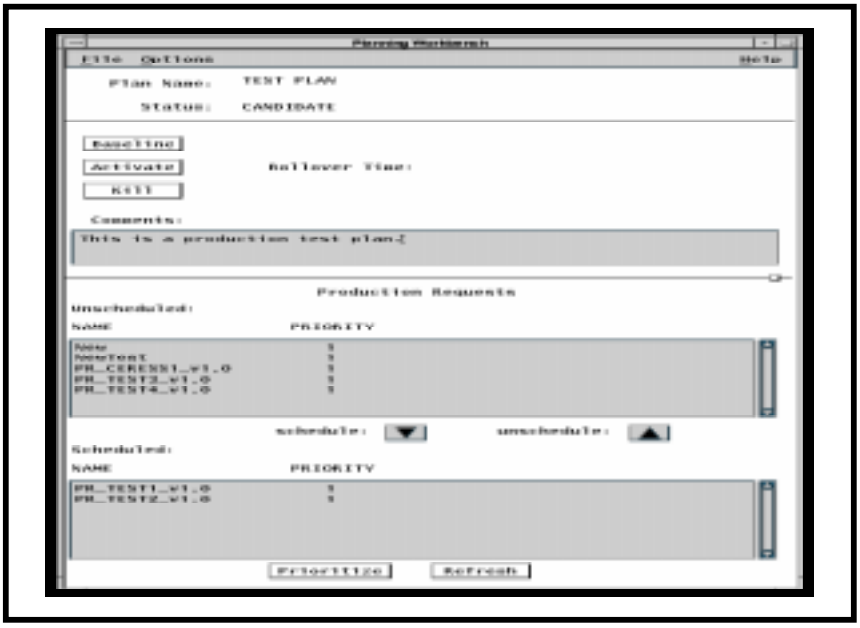


Figure 26.14.6-1. Planning Workbench GUI

Data concerning the currently active production plan are displayed.

If you want to "kill" (deactivate) the currently active production plan without activating a replacement, click on the **Kill** button.

Whenever you activate a plan (by clicking on the **Activate** button), you automatically "kill" the currently active plan.

6 Select **File** → **New** from the pull-down menu.
The "New" window appears.

7 Type a name for the new plan, then press the **Tab** key on the keyboard.

The **Planning Workbench** GUI is displayed.
The **Plan Name** is displayed.
The **Status** displayed is **Candidate**.

8 Type the desired date (in ***MM/DD/YY*** format), then press the **Tab** key on the keyboard to advance to the next field.

9 Type the desired time (in ***hh:mm*** format), then press the **Tab** key on the keyboard.
The **Rollover Time** is displayed.

10 Type any relevant comments (up to 255 characters) in the **Comments** field.

11 Move PRs between the **Unscheduled** and **Scheduled** lists as necessary by selecting (highlighting) the PR to be moved by clicking on the PR in the list from which it is to be moved then clicking on the up or down arrow button (as applicable) to move the PR to the other list. Highlighted PR disappears from one list and appears on the other.

The unscheduled and scheduled PR lists are scrollable.

12 When the **Scheduled** list accurately reflects the PRs to be scheduled in the production plan, select **File** → **Save** (or **File** → **Save As**) from the pull-down menu to save the new production plan.
The new production plan is saved.

13 If the new plan is to be activated immediately, click on the **Activate** button to activate the new plan.
The currently active plan is killed (deactivated) and the new plan is activated.
- The **Production Planning Timeline** GUI is displayed.

14 If the new production plan is to be used as a baseline plan, click on the **Baseline** button.
The "New" window appears.
The plan is recorded as well as the time of baselining so that it can be used in comparing future processing results with planned objectives.

15 If the production plan being displayed is active and should be deactivated, click on the **Kill** button. The "New" window appears.
The plan is deactivated without activating another plan.

The progress of one or more PGEs running within the PDPS may be monitored. The COTS tool used for this purpose is AutoSys® by Atria Software. Each Data Processing Request results in seven AutoSys jobs that are boxed together. An AutoSys job name follows the template:

PGEname#Suffix

where *PGEname* is replaced by the name of the PGE, and *Suffix* is a character indicating the job phase of the DPR

For example, for a scheduled PGE named MOPITT4, the AutoSys jobs making up that DPR would be:

MOPITT4#A
MOPITT4#S
MOPITT4#P
MOPITT4#E
MOPITT4#p
MOPITT4#I
MOPITT4#D

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The required servers of the ECS System are up and running.
2. A DPR has been scheduled successfully.

To monitor production, execute the procedure steps that follow:

---

1 In any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the DPS host.
- It is recommended that this procedure begin within a new command shell on a DPS Host.

- Set the DISPLAY environment variable. At the UNIX prompt on the DPS host (e.g. p0pls01), type **setenv DISPLAY** ***terminal_id***, press **Return**.

2 Set the DCE environment variable, at the UNIX prompt on the DPS host (e.g. p0pls01), type **dce_login** ***DCE_user_name DCE_password***, press **Return**.

3 Set the UNIX environment variable,at the UNIX prompt on the DPS host, change to the directory where the scripts are located (e.g. **cd /usr/ecs/TS1/CUSTOM/utilities),** then

- Type **setenv MODE** ***mode*** (e.g. TS1).
- Type **source** ***environment_setup_file*** (e.g., **EcCoEnvCsh** for C shell users).

4 At the UNIX prompt on the DPS Host, under the directory where the scripts are located (e.g. **cd /usr/ecs/TS1/CUSTOM/utilities**), type **EcDpPrAutosysStart** ***mode application_id*** **&**, then press **Return**.

- The ***mode*** is one of modes used in the ECS system (e.g., TS1).
- The ***application_id*** is a numerical number (e.g., 1).
- For example: **EcDpPrAutosysStart TS1 1, Return.**
- A GUI labeled **AutoSys** will be displayed.
- This GUI will contain eight buttons for invoking various tools available under AutoSys.

5 In the AutoSys GUI, click on the **Ops Console** button.

- A GUI labeled **AutoSys Job Activity Console** GUI will be displayed.
- The main subwindow of this GUI will contain a dynamically updated list of AutoSys jobs (seven jobs make up one DPR) currently scheduled.
- To disable dynamic updating of the main subwindow (which may be distracting), click on the **View** menu, then choose **Select Jobs**. A GUI labeled **Job Selection** will be displayed. Under the label **Select by Name**, click on the square labeled **All jobs**, then click on the **OK** button.
- DPRs will be listed in the column labeled **Job Name** and their statuses (e.g. SUCCESS) will be listed in the column labeled **Status**.
- To view job status information for a particular DPR, click on a DPR. Below the main subwindow, available job status information will be displayed.
- To view the existing event report, under the label **Reports**, click on the middle diamond labeled **Event**. The current event status for the selected DPR will be displayed in the subwindow labeled **Event Report**. Alternatively, the summary report for the selected DPR will be displayed by clicking on the middle diamond labeled **Summary**.
- To view the job definition, under the label **Show**, click on the **Job Definition** tab. A GUI labeled **Job Definition** will be displayed. The selected DPR is shown in **Job Name**. To kill the DPR, click on the **Delete** tab. ***(Warning: Be very careful to avoid deleting wrong DPR!)***
- Exit the **AutoSys Job Activity Console** by clicking on the **Exit** button.

6 In the AutoSys GUI, click on the **JobScape** button.

- A GUI labeled **JobScape** will be displayed.
- The main subwindow of this GUI will contain a dynamically updated list of AutoSys jobs (seven jobs make up one DPR) currently scheduled. The colors indicate the job statuses for DPRs and their jobs at present.

- There are eleven job statuses: **ACTIVATED, STARTING, RUNNING**, **SUCCESS, FAILURE, TERMINATED, RESTART**, **QUE_WANT**, **ON_ICE**, **OFF_HOLD,** and **INACTIVE**. The color chart is on the left of GUI.
- To view job status information for a particular DPR (or job) in detail, click on a DPR (or job), then click on the **Job Console** button. A GUI labeled **Job Console** will be displayed. The information shown here is similar to that shown in **Ops Console** as mentioned above.
- To change the status of a job for a particular DPR, click on a job under that DPR, then press the right button of the mouse. Choose one of the functions in the lower part of the menu. For example, to hold a job, choose **On Hold**, then click the **Yes** button.
- ***Suggestion:*** For a scheduled DPR, hold all jobs for that DPR except the first one (Allocation). After the first job runs successfully, release the next job (Staging) by choosing **Off Hold**. Repeat until all jobs are done. Therefore, if a job fails, it can be fixed and rerun without interfering with the others.
- To exit the **JobScape**, click on the **File** menu, then choose **Exit**.

7 In the AutoSys GUI, click on the **TimeScape** button.

- A GUI labeled **TimeScape** GUI will be displayed.
- The main subwindow labeled **Job Name** of this GUI will contain a dynamically updated list of AutoSys jobs (seven per DPR) currently scheduled.
- To disable dynamic updating of AutoSys jobs (which may be distracting), click on the **Freeze Frame** button.
- The color of each job indicates its status according to the legend on the left side of the GUI.
- The time line is shown on the right side of the GUI with time marked at the top. A red vertical line (dashed) indicates the current time.
- Exit the **TimeScape** GUI by clicking on the **File** menu and selecting **Exit**.

8 To quit AutoSys, in the AutoSys GUI, click on the **Exit** button.

- The AutoSys GUI will disappear.
- Refer to Section (Troubleshooting and General Investigation) if any job fails on AutoSys.

On workstation **x0pls##**, at the UNIX prompt in a terminal window, enter steps as in step 1 below.

NOTE: The **x** in the workstation name will be a letter designating your site: **g** = GSFC, **m** = SMC, **l** = LaRC, **e** = EDC, **n** = NSIDC, **o** = ORNL, **a** = ASF, **j** = JPL; the ## will be an identifying two-digit number (e.g.,**g0pls01** indicates a Planning and Data Processing subsystem(PLS) workstation at GSFC).

Prior to the rlogin, enter **setenv DISPLAY <local_workstation IP address>:0.0.** The **<ipaddress>** is the ip address of **x0pls##,** and **xterm** is required when entering this command on a Sun terminal.

26.14.7 Monitoring Production in PDPS Subsystem

Example of monitoring production usisng above procedures:

1 **telnet to (PDPS) p0sps06** or from the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to a PLN Host.

11 login: **ID,** password:

3 *Login to DCE* **(dce_login DCE_user_name DCE_password and then press Enter Key.)**, ***setenv DISPLAY ….:0.0***

4 Change the working directory to the location where the script for starting the Autosys is stored. (e.g**. cd /usr/ecs/<mode>/CUSTOM/utilities).**

5 **setenv MODE <MODE>**

6 **source EcCoEnvCsh**

4 At the UNIX prompt on the PLN Host, type: **EcDpPrAutosysStart <MODE> <Autosys Instance>, example: TS1 1, &**, press **Return**.

- A GUI labeled **AutoSys** will be displayed.
- This GUI will contain eight buttons for invoking various tools available under AutoSys.

5 In the AutoSys GUI, click on the **Ops Console** button.

– The AutoSys GUI will be displayed.

6 Select the desired display and view the contents.

– Selections include:

1. Autosys Job Activity Ops Console
2. HostScape
3. TimeScape
4. JobScape

- The main subwindow of this GUI will contain a dynamically updated list of AutoSys jobs (seven jobs make up one DPR) currently scheduled.
- To disable dynamic updating of the main subwindow (which may be distracting), click on **Freeze Frame** in the small subwindow labeled **Show**.
- DPRs will be listed in the column labeled **Job Name** and their statuses (SUCCESS, FAILURE, TERMINATED) will be listed in the column labeled **Status**.
- To view job status information for a particular DPR, click on a DPR. Below the main subwindow, available job status information will be displayed.
- To view the existing event report, under the label **Reports**, click on the middle diamond labeled **Event**. The current event status for the selected DPR will be displayed in the subwindow labeled **Event Report**.
- Exit the **AutoSys Job Activity Console** by clicking on the **Exit** button.

Suggestion: For a scheduled DPR, hold all jobs for that DPR except the first one (Allocation). After the first job runs successfully, release the next job (Staging) by choosing Off Hold. Repeat until all jobs are done. Therefore, if a job fails, it can be fixed and rerun without interfering with the others.

This is especially important for the Postprocessing job. If the PGE fails and the Postprocessing job is not **On Hold**, the DPR will have to be deleted and a new one created. This happens because PDPS will have deleted all references to the output granules in the PDPS database

7 In the AutoSys GUI, click on the **TimeScape** button.

- A GUI labeled **TimeScape** GUI will be displayed.
- The main subwindow labeled **Job Name** of this GUI will contain a dynamically updated list of AutoSys jobs (seven per DPR) currently scheduled.
- To disable dynamic updating of AutoSys jobs (which may be distracting), click on the **Freeze Frame** button.
- The color of each job indicates its status according to the legend on the left side of the GUI.
- The time line is shown on the right side of the GUI with time marked at the top. A red vertical line (dashed) indicates the current time.
- Exit the **TimeScape** GUI by clicking on the **File** menu and selecting **Exit**.

8 To quit AutoSys, in the AutoSys GUI, click on the **Exit** button.

7 The AutoSys GUI will disappear.

9 Use the "**tail -f em.log**" to create a permanent record of a log file if debugging is necessary. Note: the xx.log file has to be created first: "**vi em.log**"

10 To look at the Data Base type: **setenv MODE <MODE>, cd utilities**

source EcCoEnvCsh, cd dbr, dbrowser-syb <MODE> 2 &

11 Scrips to restart the servers and subsystems are located in: **/home/cmts1/restart/drop4/**

26.14.8 Using the Q/A Monitor

The Q/A Monitor allows the output products produced during a PGE run to be accessed and examined. Input test data granules and Production History files can be retrieved in the same manner. The Q/A Monitor retrieves output products based on the collection name (*i.e.* the ESDT) and time of Insertion to the Science Data Server.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow. The Q/A Monitor is discussed in more detail in Section 15.

Assumptions:

1. The required UNIX environment variables have been set properly.
2. The desired output products have been successfully Inserted to the Science Data Server.

To use the Q/A Monitor, execute the procedure steps that follow:

1 **telnet to (PDPS) p0sps06** or from the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to a PLN Host.

8. Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the PLN Host.

9. Set the DISPLAY environment variable: **setenv DISPLAY terminal_id, press Return.**
10. Type **setenv <mode> (**e.g..,**TS1).**
11. Type **source** environment_setup_file **(**e.g**., EcCoEnvCsh** for C shell users**).**

2 login: **ID,** password:

3 *Login to DCE* **(dce_login DCE_user_name DCE_password and then press Enter Key.***), **setenv DISPLAY ….:0.0***

- Change the working directory to the location where the script for starting the Autosys is stored. (e.g**. cd /usr/ecs/<mode>/CUSTOM/utilities).**

4 At the UNIX prompt on the PLN Host, type: **EcDpPrQaMonitorGUIStart <MODE> <Q/A Monitor Instance>, example: TS1 1, &**, press **Return**.

- The ***mode*** is the operations mode.
- The Q/A Monitor GUI will be displayed.
- Various messages from the Q/A Monitor will appear in this window as it is running.

5 In the Q/A Monitor subwindow labeled **Data Types,** select an ESDT from the list presented and click on it.

- Use the scroll bars if necessary to locate desired ESDT.
- Optionally, use the **Find** field and button to locate an ESDT.

6 In the Q/A Monitor, under the label **Data Granule Insert Date (mm/dd/yy)**, set the date range within which the search for granules of the ESDT selected will be conducted.

- The date range can be made arbitrarily large to select all granules of a particular collection (ESDT).
- The dates refer to date of granule Insert to the Science Data Server.

7 In the Q/A Monitor, click on the **Query** button.

- The results of the query will be displayed in the bottom window, labeled **Data Granules**.
- All granules having the ESDT selected in step 3 and having Insert times within the date range specified in step 4 will be listed in this window.

8 In the Q/A Monitor subwindow labeled **Data Granules**, click on one of the data granules listed to be examined.

- The data granule selected will be highlighted.

9 To retrieve the data granule's Production History file, click on the **Retrieve Prod History** button.

- The Production History (PH) tar file corresponding to the selected data granule will be retrieved from the Science Data Server and placed on the local machine (a PLN Host) in the directory /var/tmp.
- The PH can then be moved or copied manually from the /var/tmp directory to a user working directory for examination.
- Only the PH file is retrieved with the **Retrieve Prod History** button.
- If only the PH is desired, exit this procedure. To retrieve the data granule itself, continue on to step 8.

10 To retrieve the data granule, click on the **Retrieve Data Granule** button and note its file name (listed in the entry for the granule; you may have to scroll over to the right to see it).

- The data granule selected will be retrieved from the Science Data Server and placed on the local machine (a PLS Host) in the directory /var/tmp.
- A granule of any format (binary, ASCII, HDF, HDF-EOS) may be retrieved in this manner. Only HDF and HDF-EOS granules, however, may be further visualized using EOSView as described in the next steps.

11 To examine the data granule, click on the **Visualize data** tab.

- The **Visualize data** page will be displayed.

---

12 In the main subwindow on the **Visualize data** page, locate the file name of the granule retrieved in step 8 and click on it.

- The item selected will be highlighted and will appear in the **Selection** subwindow below.

13 Click on the **Visualize** button.

- This action will invoke **EOSView** with the granule selected.
- The granule must be HDF or HDF-EOS format.

14 When tasks are completed with the Q/A Monitor GUI, click on the **File** menu, then choose **Exit**.

26.15 Postprocessing and General Investigation

An important part of SSI&T is verifying that the output files produced at the DAAC are identical (within particular tolerances) to the test output files delivered with the DAPs. A successful comparison is a strong indication that the porting of the science software from the development facility at the SCF to the operational facility at the DAAC has not introduced any errors.

A number of file comparison tools are available during SSI&T via the SSIT Manager GUI or they can be invoked from the UNIX command line. Two tools are available for comparing HDF or HDF-EOS files, one tool for comparing ASCII files, and another tool for assisting in comparing binary files.

It is assumed that the Instrument Team has delivered test output files (produced at their SCF) with which to perform the comparison.

26.15.1 Examining PGE Log Files

Three log files are produced by PGEs during runtime: the Status log, User Log, and the Report log. These log files are written by the SDP Toolkit and by the science software using the Toolkit's Status Message Facility (SMF). The location of these log files is specified in the Process Control File (PCF). When the PGE is built and run with the SCF version of the SDP Toolkit, the location and file names of the log files can be set as desired. When the PGE is built

with the DAAC version of the SDP Toolkit and run within the PDPS, the location and file names of the log files is set by the system in the instantiated PCF.

The Status log file captures all error and status information. The User log file captures a subset of messages which are more informational. The Report log file captures arbitrary message strings sent by the PGE.

The section aforementioned describes how to examine log files produced by PGEs that have been built with the SCF version of the SDP Toolkit and run from the command line.

The section aforementioned describes how to examine log files (within the Production History) produced by PGEs that have been built with the DAAC version of the SDP Toolkit and run within the PDPS.

26.15.1.1 Log Files From PGEs Run Outside of the PDPS

When the PGE is run outside of the PDPS, the PCF specifies the location and file names of the log files produced. This procedure describes how to locate that information from the PCF and use it to examine the log files.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The PGE has been successfully built with the SCF version of the SDP Toolkit.
2. The PGE's PCF has been updated properly for the DAAC environment .

1 At the UNIX prompt on an AIT Sun or on the SPR SGI, type **cd *PCFpathname***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the PCF used by the PGE for which log files are to be examined.

2 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **vi *PCFfilename***, press **Return**.

- The ***PCFfilename*** is the file name of the PCF used by the PGE for which log files are to be examined.
- This brings up the file named ***PCFfilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs MOD35.pcf**, press **Return**.

3 In the editor, search for logical IDs (beginning in the first column) **10100**, **10101**, and **10102**. These are the PCF entries for the LogStatus, LogReport, and LogUser respectively. For each, note the file names in field 2 and the path names in field 3. Then quit the editor.

- If field 3 is blank, then the location is given by the default location specified in a line above the entries beginning with the "!" character.

4 At the UNIX prompt on the SPR SGI, type **vi *StatusLogPathname*/*filename***, press **Return**.

- The ***StatusLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10100. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs /PGE/MOD35/LogStatus**, press **Return**.

5 At the UNIX prompt on the SPR SGI, type **vi *UserLogPathname/filename***, press **Return**.

- The ***UserLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10101. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs /PGE/MOD35/LogUser**, press **Return**.

6 At the UNIX prompt on the SPR SGI, type **vi *ReportLogPathname/filename***, press **Return**.

- The ***ReportLogPathname/filename*** is the full path name and file name of the Status log file noted in step 3 associated with PCF logical ID 10102. When finished, quit the editor.
- Note any anomalous messages in file.
- Any text editor may be used such as *emacs*. For example, **/PGE/MOD35/LogReport**, press **Return**.

26.15.1.2 Production History Log Files From PGEs Run Within the PDPS

The Production History (PH) is created during PGE execution within the PDPS and then Inserted into the Data Server upon PGE completion. The PH is a UNIX tar file that includes the PGE log files.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The PDPS archive configuration area is properly set up.
2. The environment variable DataServer contains the full path name to the archive. This is typically /imf/archive/ or /imf_data/archive/ and varies at each DAAC.

1 At the UNIX prompt on the SPR SGI, type **cd $DataServer/PH**, press **Return**.

- The **$DataServer** is an environment variable containing the full path name of the Data Server archive and **PH** is a subdirectory under **$DataServer** containing the Production History tar files.
- For example, type **cd $DataServer/PH**, press **Return**.

2 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **ls -al**, press **Return**.

- A list of the current contents will be displayed. These will be Production History tar files.

- The file names of PH files are named *PGEname#versionMMDDYYhhmm*_runtime.tar_*UR* where *PGEname* is replaced by the PGE name, *version* is replaced by the PGE version, *MMDDYY* is replaced by the Insert date in the month-day-year format, *hhmm* is replaced by the Insert time in the hours-minutes format, and *UR* is replaced by the Universal Reference. For example, the PH file name for Release 4.1 of a SAGE III PGE named sage_1t Inserted on December 14, 1999 at 2:00pm could be: sage_1t#2.11214991400_runtime.tar_YAAa005Li_19991214135815.
- Look for the PH of interest.

3 At a UNIX prompt on the AIT Sun or on the SPR SGI, type **cp *PHtarFilename WorkingPathname***, press **Return**.

- The ***PHtarFilename*** is the file name of the Production History tar file.
- The ***WorkingPathname*** is the full path name to some working directory in which the Production History tar file is to be placed and examined.

4 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **cd *WorkingPathname***, press **Return**.

- The ***WorkingPathname*** is the full path name to the working directory specified in step 3.

5 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **tar xvf *PHtarFilename***, press **Return**.

- The ***PHtarFilename*** is the file name of the Production History tar file in the working directory.
- This command will untar the Production History tar file, extracting its component files into the current directory.

6 At the UNIX prompt on the AIT Sun or on the SPR SGI, type vi StatusLogFilename, press Return.

- The StatusLogFilename is the file name of the Status log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as emacs. For example, emacs LogStatus, press Return.

7 At the UNIX prompt on the AIT Sun or on the SPR SGI, type vi UserLogFilename, press Return.

- The UserLogFilename is the file name of the User log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as emacs. For example, emacs LogUser, press Return.

8 At the UNIX prompt on the AIT Sun or on the SPR SGI, type vi ReportLogFilename, press Return.

- The ReportLogFilename is the file name of the Report log file within the PH. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as emacs. For example, emacs LogReport, press Return.

26.15.1.3 History Log Files From Failed PGEs Run Within the PDPS

The History Log(HL) is created during PGE execution within the PDPS and then Inserted into the Data Server upon failure of the PGE. The HL is a UNIX file that includes the PGE log files, the PCF and PH.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The PDPS archive configuration area is properly set up.
2. The environment variable DataServer contains the full path name to the archive.

1 At the UNIX prompt on the SPR SGI, type cd $DataServer/FAILPGE, press Return.
- The $DataServer is an environment variable containing the full path name of the IMF Data Server archive and FAILPGE is a subdirectory under $DataServer containing the History Log files.
- For example, type cd $DataServer/FAILPGE, press Return.

2 At the UNIX prompt on the SPR SGI, type **ls -al**, press **Return**.
- A list of the current contents will be displayed. These will be History Log files.
- The file names of HL files are named *PGEname#versionMMDDYYhhmm*_runtime.tar_*UR* where *PGEname* is replaced by the PGE name, *version* is replaced by the PGE version, *MMDDYY* is replaced by the Insert date in the month-day-year format, *hhmm* is replaced by the Insert time in the hours-minutes format, and *UR* is replaced by the Universal Reference. For example, the HL file name for Release 4.1 of a SAGE III PGE named sage_1t Inserted on December 14, 1999 at 2:00pm could be: sage_1t#2.11214991400_runtime.tar_YAAa005Li_19991214135815.
- Look for the HL of interest.

3 At a UNIX prompt on the SPR SGI, type **cp *HLFilename WorkingPathname***, press **Return**.
- The ***HLFilename*** is the file name of the History Log file.
- The ***WorkingPathname*** is the full path name to some working directory in which the History Log file is to be placed and examined.

4 At the UNIX prompt on the SPR SGI, type **cd *WorkingPathname***, press **Return**.
- The ***WorkingPathname*** is the full path name to the working directory specified in step 3.

5 At the UNIX prompt on the SPR SGI, type **vi *HLFilename***, press **Return**.
- The ***HLFilename*** is the file name of the History Log. When finished, quit the editor.
- Note any error or warning messages in file.
- Any text editor may be used such as *emacs*. For example, **emacs *HLFilename***, press **Return**.

26.15.2 The Production History

The Production History (PH) is a UNIX tar file that is produced and archived for every run of a PGE in the PDPS. Each PH can be uniquely retrieved from the Data Server.

PH files are located in the $DataServer/PH directory and have the following naming convention:

*PGEname#versionMMDDYYhhmm*_runtime.tar_*UR*

where *PGEname* is replaced by the PGE name, *version* is replaced by the PGE version, *MMDDYY* is replaced by the Insert date in the month-day-year format, *hhmm* is replaced by the Insert time in the hours-minutes format, and *UR* is replaced by the Universal Reference.

For example, a PH file name for Release 4.1 of a SAGE III PGE named sage_1t Inserted on December 14, 1999 at 2:00pm would be:

sage_1t#2.11214991400_runtime.tar_YAAa005Li_19991214135815.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The PDPS archive configuration area is properly set up.
2. The environment variable DataServer contains the full path name to the archive.

1 At the UNIX prompt on an AIT Sun or on the SPR SGI, type **cd $DataServer/PH**, press **Return**.

- The **$DataServer** is an environment variable containing the full path name of the IMF Data Server archive and **PH** is a subdirectory under **$DataServer** containing the Production History tar files.
- For example, type **cd $DataServer/PH**, press **Return**.

2 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **ls -al**, press **Return**.

- A list of the current contents will be displayed. These will be Production History tar files.
- The file names of PH files are named *PGEname#versionMMDDYYhhmm*_runtime.tar_*UR* where *PGEname* is replaced by the PGE name, *version* is replaced by the PGE version, *MMDDYY* is replaced by the Insert date in the month-day-year format, *hhmm* is replaced by the Insert time in the hours-minutes format, and *UR* is replaced by the Universal Reference. For example, the PH file name for Release 4.1 of a SAGE III PGE named sage_1t Inserted on December 14, 1999 at 2:00pm could be:sage_1t#2.11214991400_runtime.tar_YAAa005Li_19991214135815.
- Look for the PH of interest.

3 At a UNIX prompt on the AIT Sun or on the SPR SGI, type **cp *PHtarFilename WorkingPathname***, press **Return**.

- The ***PHtarFilename*** is the file name of the Production History tar file.
- The ***WorkingPathname*** is the full path name to some working directory in which the Production History tar file is to be placed and examined.

4 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **cd *WorkingPathname***, press **Return**.

- The ***WorkingPathname*** is the full path name to the working directory specified in step 3.

5 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **tar xvf *PHtarFilename***, press **Return**.

- The ***PHtarFilename*** is the file name of the Production History tar file in the working directory.
- This command will untar the Production History tar file, extracting its component files into the current directory.

6 At the UNIX prompt on the AIT Sun or on the SPR SGI, type **vi *PHcomponentFile***, press **Return**.

- The ***PHcomponentFile*** is the file name of one of the PH component files. The component files are:
  - *PGEname#versionMMDDYYhhmm*.Log - Contains the DPR ID, the actual command used to run the PGE, resource usage information, the PGE exit status, and files used by the PGE.
  - *PGEname#versionMMDDYYhhmm*.Pcf - The actual instantiated PCF used when running the PGE.
  - *PGEname#versionMMDDYYhhmm*.ProdLog - Contains the DPR ID, the PGE ID, and resource usage information (same as in the .Log file).
  - *PGEname#versionMMDDYYhhmm*.Profile - Contains the environment variables defined during the execution of the PGE including the contents of the PATH environment variable.
  - *PGEname#versionMMDDYYhhmm*.TkReport - The Report log file, same as is produced when run outside of the PDPS.
  - *PGEname#versionMMDDYYhhmm*.TkStatus - The Status log file, same as is produced when run outside of the PDPS.
  - *PGEname#versionMMDDYYhhmm*.TkUser - The User log file, same as is produced when run outside of the PDPS.
  - *ESDTmmddyyHHMM*.met - All target MCFs for all Inserts on behalf of the PGE. The *ESDT* is the ESDT ShortName into which the file was Inserted, *mmddyy* is the month, day, and year of the Insert and *HHMM* is the time of the Insert.

26.16 Examining PDPS-Related Scripts and Message Files

This section describes how users may access files, in addition to the PGE-produced log files, which are created during the execution of a DPR job and which may hold information useful in tracing processing problems.

Some of these files are written by default to directory paths that can only be accessed on either the SGI processor machine or one of the Sun workstations. More detailed descriptions of these files and the conditions under which they are generated will be supplied in future Green Book versions.

26.16.1 Examining AutoSys JIL Scripts

JILxxxxxxxxx is the Job Information Language (JIL) script that defines the DPR job to **AutoSys** and which must be submitted to the **AutoSys** Database before a DPR job can be run. The name of the file created is system-generated and begins with the characters 'JIL' followed by nine characters (e.g. JILAAAa0066c).

Sample file content:

```
insert_job: 5251_823122483_1
job_type: command
command:  /usr/ecs/{mode}/CUSTOM/data/bin/sgi/EcDpAtExecutionMain
5251_823122483_1
machine: spr1sgigsfc
std_out_file: /home/cboettch/mockpge_msfc/out/dpat_std.out
std_err_file: /home/cboettch/mockpge_msfc/out/dpat_std.err
profile: /usr/ecs/<MODE>/CUSTOM/data/bin/sgi/EcDpAtRunProfile.sh
```

To examine JILxxxxxxxx scripts on the AIT Sun, execute the procedure steps that follow:

1 At the UNIX prompt on an AIT Sun, type **cd *JILscriptPathname***, press **Return**.

- The ***JILscriptPathname*** is the full path name to the location of the JILxxxxxxxx scripts to be examined.

2 At the UNIX prompt on the AIT Sun, type **vi *JILscriptFilename***, press **Return**.

- The ***JILscriptFilename*** is the file name of the JILxxxxxxxx script to be examined.
- This brings up the file named ***JILscriptFilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs *JILscriptFilename***, press **Return**.

26.16.2 Examining Application Log Files (ALOG)

Most of the custom code used during SSI&T routinely produce log files. For example, the SSIT Manager produces a log file named **EcDpAtMgr.log** and the tool used to Insert SSEPs to the Data Server (EcDpAtInsertExeTarFile.sh) produces a log file named **EcDpAtInsertExeTarFile.log.** These files are placed in the directory in which the tool was executed. If the **SSIT Manager** is run from the user's home directory, then the log files for each of the associated tools will be found in the user's home directory. Log files are produced at the first invocation of the tools, even if no messages are written to them. During subsequent use of the tools, the associated log files will be appended.

Log files are generally named according to the convention:

ApplicationName.log

where *ApplicationName* is replaced with the name of the tool's executable binary. For tools that are shell scripts (*e.g.* .sh files), the shell name is left out of the log file name. For example, the tool EcDpAtInsertStaticFile.sh produces a log file named **EcDpAtInsertStaticFile.log** and not EcDpAtInsertStaticFile.sh.log.

Where an **SSIT Manager** application has been run using login **cmts1**, pw: **ecsu$er** with **dce_login**,the log files will be found using path: **/usr/ecs/{MODE}/CUSTOM/logs/.**

DCE failures have been encountered when installing **ESDT's , MCF's** and **.met files**. The term bounce the servers has been widely used in conjunction with the effort to re-install or delete files. **Bounce** means to **shut down a server** and then **bring them back up** to rid the servers of unwanted or old bindings. The nature of what needs to be done is outlined as follows:

1 Install or Delete **ESDT's** - the **SDSRV** and **ADSRV** need to be bounced after installation or removal of **ESDT**'s to allow for a refresh of the **DCE** cell management.

2 For **PGE…..odl, MCF's** and **.met files,** bouncing the servers **SDSRV** and A**DSRV** need to done after installation and reinstallation.

3 This can be done by loging into **ECS Assistant** for each server. The login should be with generic **ID**: and **PW**:**,** and **dce_login DCE_user_name DCE_password and then press Enter Key.**.

26.17 PDPS Troubleshooting - The PGE Job has Failed

- The PGE Job has failed, but the DPR has not gone into "Failed-PGE" processing
- The Post-Execute Job has failed
- The PGE Job and Post-Execute Job have both failed, but the DPR has not gone into "Failed-PGE" processing
- The PGE Job has failed and the DPR has gone into "Failed-PGE" processing

26.17.1 The PGE Job has failed

This condition is indicated when the PGE job only is red in **AutoSys**. This is hard to do, because the **AutoSys** job definition for this job says to allow any exit code to indicate success. This is because we want the next job, the post-execute job, to continue
even if this job fails. This job will "succeed" even if the PGE Wrapper job, **EcDpPrRunPGE,** doesn't exist. This job can fail if **AutoSys** cannot see the machine machine.

26.17.2 The Post-Execute Job has failed

This condition is indicated when the Post-Execute Job only is red in **AutoSys**. This happens when the PGE job never ran or if for some other reason (such as a mount point problem) the Execution Manager job cannot read the log file created by **EcDpPrRunPGE.**
Check that **/usr/ecs/B302TS1/CUSTOM/bin/DPS/EcDpPrRunPGE** and **EcDpPrRusage** exist on the science processor and that they are not links.
Check that **/usr/ecs/DEV04/CUSTOM/data/DPS** on the science processor is mounted or linked to **/usr/ecs/DEV04/CUSTOM/data/DPS** on the queuing server machine.

26.17.3 The PGE Job and Post-Execute Job have both failed

This condition is indicated when both the PGE and Post-Execute Jobs are red in **AutoSys**, but no other jobs are red. This indicates that the Post-Execute job has read the log file created by **EcDpPrRunPGE** in the runtime directory and has found an exit status not equal to 0. However, it failed to de-stage the failed **pge tar file**.

26.17.4 The PGE Job has failed and the DPR has gone into "Failed-PGE" processing

This condition is indicated when the entire job box has turned red along with post-execute, de-staging and de-allocation jobs. A Failed PGE Tar File has been created and archived.
A PGE may fail for many reasons. Some of the possible causes are documented here:
11 The PGE is the wrong architecture.
This happens when the PGE was miss-defined as **New32, Old32** or **64** from the SSIT Operational Metadata GUI. The PGE will core dump because of this problem. To fix this you need to go back to **the SSIT Operational Metadata GUI** and enter the correct architecture, then delete any **DPRs** created for that **PGE** and recreate them.

- One of the expected inputs for the PGE is missing

The first reason for this is that an expected input of the PGE is NOT defined in the **PGE ODL**. Check the error messages for something about a missing Logical Id and then check the **PGE ODL** for the expected **Logical Id.**

This can also happen when a miscommunication causes Subscription Manager to release a PGE despite it missing one (or more inputs). To find out if this is the case, verify that all inputs to the DPR have their availability flag set to 1 in PlDataGranuleShort and the corresponding entries in **PlDprData** have the accepted field set to 1.

- The leapseconds file is incorrect.

26.17.5 PDPS Troubleshooting - A single DPS job has failed or is hanging

The entire Job Box is hung

A DPS Allocation job is hanging

A DPS Allocation job has failed

A DPS Staging job is hanging
A DPS Staging job has failed
A DPS Preprocessing job is hanging
A DPS Preprocessing job has failed
A DPS PGE job is hanging
A DPS PGE job has failed
A DPS Post-processing job has failed
A DPS De-staging job is hanging
A DPS De-staging job has failed
A DPS De-allocation job has failed
All known problems are corrected and the re-start of the job fails again

Non-PDPS servers are down

Back to the PDPS Troubleshooting home page

26.17.5.1 The entire Job Box is Hung

This condition is determined by noting that the entire Job Box (all 8 job steps) are the same color, and it is either the one indicated for "Inactive" jobs or the one for "On Hold" jobs. Check the legend to the left on the Jobscape display for the job box color meanings.

- The AutoSys Event server or one of the AutoSys clients could be down. See
- Attempt to re-start the Job Box by selecting the top of the Job Box.

26.17.5.2 A DPS Allocation job is hanging

This condition is determined by noting that the Allocation job has turned green to indicate that it is running, but that it never turns red (failed) or blue (success). If you "tail" the DPR .err file (eg: "tail -f **/usr/ecs/OPS/CUSTOM/logs/MODPGE08#s28035000OPS.err"**) you see that nothing is happening, or that the job is in a retry loop.

- The Science Data Server (SDSRV) may be waiting for a request to Data Distribution (DDIST) to distribute the PGE tar file, but it can't because Storage Management (STMGT) is down. Go to where the DDIST GUI is running. Refresh the GUI. Check to see if the requestor source is EcDpPrEM and that the state is "Suspended with Errors". If this is the case, then you will have to bounce STMGT. After this is done, select the request on the DDIST GUI and click on "resume".

- The Science Data Server (SDSRV) may be waiting for a request to Data Distribution (DDIST) to distribute the PGE tar file, but it can't because Storage Management can't FTP the file to the data directory on the science processor disk. Go to where the DDIST GUI is running. Refresh the GUI. Check to see if the requestor source is EcDpPrEM and that the state is "Suspended with Errors". Does the target directory exist? Can you FTP a file to the directory on the science processor? If the answer to these questions is no, then fix the problem, and resume the request.

- If you observe that the Allocation Job is in a retry loop, then the SDSRV may be down. See Non-PDPS servers are down. Note that the first retry is designed to fail, because the software is retrieving server-side information to refresh the client-side at this point.

- The request may be waiting on the archive to stage the file. If there are several other requests in progress, the PGE acquire may have to wait until one or more of them completes. Check the state in the DDIST GUI - if it is in "staging" state, then the request should eventually complete.

26.17.5.3 A DPS Allocation job has failed

This condition is determined by noting that the Allocation job has turned red.

- Look at the .ALOG file (in **/usr/ecs/{MODE}/CUSTOM/logs**). If it is there, then look for the following:

  A message of "Error: unable to update Machine in Autosys" means that DPS is unable to access the AutoSys database. The auto.profile **in /usr/ecs/MODE/CUSTOM/bin/DPS** has the wrong settings for AUTOSYS and AUTOUSER parameters. Although they may differ from DAAC to DAAC, the expected values are:

  AUTOSYS = /usr/ecs/MODE/COTS/autotreeb/autosys
  AUTOUSER = /usr/ecs/MODE/COTS/autotreeb/autouser

- To fix the problem, you either need to run the AutoSys Mkcfg again or go into the auto.profile file and change the values by hand.
- A message of "Unable to determine type of UR" means that the PGE tar file has not been inserted. To verify this is the problem check the PlResourceRequirement table in the PDPS database. There should be a non-null entry for the field exeTarUR. If that field is null, you need to go back to the SSIT procedure and insert the EXE Tar File. Then you should be able to re-start the job and watch it complete successfully.

If the .ALOG file is NOT present. Then do the following:

- Bring up the Autosys Ops Console and select the Allocation Job that has failed.
- Check the return code. A value of 122 means that owner of the job DOES NOT HAVE WRITE PERMISSION to the log files directory. You need to find out the user account that was used to bring up Autosys and verify that it is correct and should have write permission to the logs directory.

26.17.5.4 A DPS Staging job is hanging

See A DPS Allocation job is hanging

26.17.5.5 A DPS Staging job has failed

This condition is determined by noting that the Staging job has turned red. Look at the .ALOG file (**in /usr/ecs/{MODE}/CUSTOM/logs**) for the DPRID of the job that has failed.

- A message of "ESDT Acquire Failed for UR...." means that SDSRV had trouble processing one of the acquire requests. In this case re-starting the job should allow the acquire to succeed.

26.17.5.6 A DPS PreProcess job has failed

This condition is determined by noting that the Pre-Process job has turned red.

Look at the .ALOG file (in **/usr/ecs/{MODE}/CUSTOM/logs**) for the DPRID of the job that has failed. If it is there, then look for the following:

- A message of "NOFREECPUS" means that all of the Science Processor CPUs are busy and the PreProcess job went through its maximum number of retries to find an available CPU. You can just start the job again and it will work its way through its retries until a CPU is available.

Possible reasons for a job to run out of CPU resources:

- PGEs are taking longer to run than expected. DPS plans for execution times specified during SSIT, and if those times are exceeded by a large margin (by an executing PGE) it is possible that a PGE that is "ready to run" will be CPU starved.

26.17.5.7 A DPS PGE job is hung

This condition is determined by noting that the Execution job has turned orange or ossilates between orange and green.

- The AutoSys client is most likely down. See Checking the Status of AutoSys for how to verify AutoSys is up and happy.

26.17.5.8 A DPS PGE job has failed

This condition is determined by noting that the Execution job has turned red or the entire job box has turned red (failedPGE scenario).

See Troubleshooting - The PGE Job has Failed.

26.17.5.8 A DPS De-staging job has failed

The destaging job icon on the JobScape GUI will have turned red. Look at the .err log file.

- Typically, you will see a message such as "Error archiving metadata into catalog". You may also see some warning messages in the returned GlParameter list. You can disregard the warnings. If the problem occurred for an existing ESDT which has previously worked within the past day or two, then most likely STMGT is the culprit. Have someone from STMGT look at their log files, paying particular attention to changes/defects in their stored procedures.

- If you see the "Error archiving metadata into catalog" message and the ESDT is new or has recently been installed, then look at the .MCF file in the runtime directory. Get somebody from SDSRV to help you compare the values of the mandatory parameters in the metadata file with "valids" from the SDSRV database.

- "Error archiving metadata into catalog" may also be associated with a SDSRV temporary directory getting filled up.

  A message that indicates "Error archiving files" means that SDSRV is having trouble getting Storage Management to place the file(s) in the archive. Contact a Storage Management person:

- to Verify that the Archive (AMASS) is up and functional.
- Check through their logs to see why the request for archiving failed.

  It is possible that the mount point between the science processor and the Storage Management machine has been lost. Check to make sure that the file that is being de-staged can be seen on the Storage Management machine. The typical path for the mount point is: Name**}**
  /usr/ecs/{MODE}/CUSTOM/pdps/{science processor name}/data/DpPrRm/{Disk

- A message that indicates "Error modifying file usage" means that the numberOfUsage column in DpPrFile for a particular file is at 0 and the software is trying to decrement it. This column is an increment/decrement counter and is not normally decremented more times than it is incremented when under software control. However, if someone manually changes the database then the value may get out of sync and need to be manually reset to 1.

- If you see science data files in the disk partition, but no metadata files, then DDIST/STMGT is okay and SDSRV is not okay. Otherwise, suspect STMGT.

- When the problem is corrected, re-start the job from AutoSys.

26.17.5.9 All known problems are corrected and the re-start fails again

The retry information in DpPrRpcID may now be out of sync between one or more servers. Find the appropriate entry in the table by inspecting the readableTag column and remove the entry before trying to re-start the job again.

26.17.5.10 Non-PDPS servers are down

Always verify that the Science Data Server, Storage Management Servers and Data Distribution Servers are up.

1. Bring up ECS ASSIST
2. Select the correct mode
3. Click on the "monitor" button
4. Click on "cdsping all servers"
5. Observe that the status for the following servers is "Listening":
   EcDmDictServer
   EcDpPrDeletion
   EcDpPrJobMgmt
   EcDsDistributionServer
   EcDsScienceDataServer (all instances)
   EcDsStArchiveServer
   EcDsStFtpDisServer
   EcDsStPullMonitorServer
   EcDsStStagingDiskServer

   EcDsStStagingMontitorServer

   EcIoAdServer
   EcPlSubMgr
   EcSbEventServer
   EcSbSubServer
6. If any of these servers are down, contact the MSS operator to bring them up.

26.17.6 PDPS Troubleshooting - Job Activation Fails from the Planning Workbench

Error reported is "DPR Validation Failed"

1. Check to make sure that Performance data has been entered for the PGE
   1. Use the database browser or isql to get access to the PDPS database.
   2. Look at the entries for PlPerformance.
   3. For the PGE(s) that are schedule, verify there is a non-zero value for the entries in this table.
   4. If entries are 0, then run the SSIT Operational Metadata GUI to enter correct performance values.
   5. Delete the DPRs and then re-create them. Activation will succeed on the next attempt.

26.17.6.1 PDPS Troubleshooting - Input Data Problems

General description of the problem:

- The Production Request fails due to too many granules
- The Staging Job has failed due to too many granules

A Failure Due to Too Many Granules

A failure of this sort is caused by too many granules meeting the criteria for input granules for a particular DPR. At PGE registration, the number of granules we expect for each input ESDT is defined. We define the minimum number and the maximum number of granules we expect. If the number of granules found is not between the minimum and maximum number, the request fails. This will fail either during production request time, or when the PCF File is generated (Autosys PreProcessing step).

The Production Request fails due to too many granules

The Production Request fails. The ALOG displays an error message as follows:
Msg: PlPge::GetInputForDpr - Extr input to process DPR MoPGE01#2007081600OPS, for data type id MOD000#001, with logical id 599001. PlDataTypeReq has a scienceGroup of for this datatype. Expected 2 max inputs, but got 3. Priority : 2
Time : 07/09/99 17:10:52

The problem in this case was that the Production Request Editor queried the PDPS database for granules that would satisfy the data needs for this DPR and found 3 granules instead of 2 like it expected. When we inspected the PDPS database (PlDataGranuleShort) we found only two entries that satisfied the data needs for this DPR. After some investigation, the true problem was discovered.

The DPR was for 16:00:00 - 18:00:00. This particular DPR takes in the current input granule (16:00:00-18:00:00) and the previous input granule (14:00:00- 16:00:00). When it inspects the PDPS Database for data granules, it "pads" the timeframe with a (5A) configurable percentage or (4PY) a hardcoded percentage - 50%. This was 4PY, so the PRE added 50% of the granule's expected time to each side of the time-range of the granules. This means that the PRE queried the PDPS database for granules that had start and stop times within 15:00:00 - 19:00:00. In this case, there was an invalid data granule with a start time of 15:30:00 - 15:59:59. This granule was found during the query and caused the PRE to find one too many granules and fail the PR.

The Workaround:
Delete the offending granule and recreate your production request.

The Fix:

For 5A, the padding that is added to each side of the range for the query is configurable. If the padding is decreased, these kinds of granules will not be found. In addition, there is a minimum size that a granule must be before the Planning system will recognize it as a valid granule. This is a percentage, and is hardcoded in 4PY to 20% (or so), but this is configurable in 5A. So in 5A we can set this value to 50% which will filter out all granules less than 1 hour long, and we can set the padding to 50% which will filter out granules that do not start before 15:00:00. Since our granules must be an hour long, and we can assume they do not overlap, we would not get any invalid granules such as this one.

The PreProcessing Job has Failed due to Too Many Granules
This condition is indicated when the Staging Job is red in AutoSys. This happens due to the same condition as noted above with the PRE, but the data granules came in after the Production Reqeust was generated. The error message appears in the PGE log files and comlains about not being able to generate the PCF file due to too many granules for a particular logical id.

The Workaround:
Delete the offending granule and the production request and recreate the production request.

The Fix:
See fix for the previous scenario.

26.17.6.2 PDPS Troubleshooting - Jobs are activated, but do not get started in AutoSys

The Job Management Server is down
The DPR is waiting in the AutoSys queue (never got released)
Subscription Server Problems
The DPR was released but failed due to a JIL failure
The DPR was released but failed due to a AutoSys ID failure
The DPR was released but failed to be received by Job Management Server
AutoSys is not functional
AutoSys is full

The Job Management Server is down

1. Bring up ECS ASSIST
2. Select the correct mode
3. Select DPS for the subsystem
4. Click on the "monitor" button
5. Observe that the status for the EcDpPrJobMgmt server is "UP"
6. If it is "down", then do steps 7 - 8

7.Click on the "start" button
8.Repeat steps 1-5

The DPR is waiting in the AutoSys queue (never got released)

The Job Management server may have never received a ReleaseDprJob command from the PLS Subscription Manager

1.Check the database table, DpPrCreationQueue, to see if the job is still waiting to into AutoSys. If it's in this table then it probably never got a ReleaseDprJob command get from the PLS Subscription Manager, unless AutoSys is full)
2.Check the Job Management error log file to see if the ReleaseDprJob command was sent
3.If it was, then there may have been a JIL (AutoSys Job Information Language) processor problem -- a JIL FAILURE
4.If you can't find any evidence that the command was sent to Job Management, then the PLS Subscription Manager didn't send the ReleaseDprJob command. It won't send this command if it doesn't think all of the DPR's required inputs have been received. Verify this for yourself as follows:

a. For regular DPRs (ie. one without optional inputs), check to see if all of the required inputs are present. First look at the PlDprData table and find all of the granule Ids with an ioFlag of 0 (an input granule) for this DPR. Then look at the UR column for each granule Id in PlDataGranule. If all of the input granules have URs (as opposed to granule Ids), then the Subscription Manager should have sent a ReleaseDprJob command to Job Management.

Look at the Subscription Manager log file to verify that it never, in fact, sent the ReleaseDprJob command.

b.While you're looking at the Subscription Manager log file, check to see if it got a subscription notification from the

Subscription Server for any dynamic data that the DPR needs. If you believe that all of the necessary input files for
the DPR have been inserted by another DPR, then there may be Subscription Server Problems
c.If there are no Subscription Server Problems, all of the input granules for the DPR have URs and/or Subscription Manager received notification for all dynamic granules, then something may be wrong with the Subscription Manager. Check with somebody in the PLS subsystem.

The DPR was released but failed due to a JIL failure

A "JIL Failure" means that the Job Management Server had some problem placing the DPR in AutoSys. The Job Interface Language processor rejected the create job command sent to it by the Job Management Server. If you look at the completionState column for the DPR in the PDPS PlDataProcessingRequest table, you will see "JIL_FAILUR". There are 2 main reasons for this:

1.There is already a job with an identical name in AutoSys. Check this by going to the Ops Console, select View->Select Jobs and type a portion of the job name in the "Job Name" box, bracketed by the "*" or "%" wildcard character. If the job is already in AutoSys, it must be removed by using the Production Request Editor or by using the Job Management Server Client tool (selectable from the Ops Console). Never delete a job from AutoSys using the job definition GUI. This will corrupt the PDPS database.
2. The event processor is down - AutoSys is not functional.
3. The job had a problem when it was loaded into AutoSys and a malformed or mutant job box is the result. This is a job box which will stay dark blue (meaning that it was not activated) and will be missing one of the seven job steps. To correct this problem you must do the following:

- Delete the job from AutoSys by hand. To do this select the job from JobScape and right click. Select the Job Definition and then select Delete from the pop-up window. In general, it is bad practice to delete a job from Autosys using the Job Definition GUI. This can cause corruption in the PDPS database. But for this problem there is no other solution.
- Update the completionStatus of the DPR in the PDPS database for which the mutant job box was created. Youmust do this via isql and set the completionStatus = NULL (using the isql update command).
- Delete the DPR that maps to the job via the Production Request Editor. Note that you do not want to delete the entire Production Request, only the DPR that had the mutant Job Box. Any DPRs that depend on this DPR will also have to be deleted.
- Re-create this DPR and any subsequent DPRs via the Production Request Editor.

The DPR was released but failed due to a AutoSys ID failure

An "AutoSys ID" failure is indicated if the following messages appear in the Job Management ALOG file:

```
PID : 7668:MsgLink :0 meaningfulname :DpPrAutosysMapList::GetAutosysIDByDpr
 Msg: unable to find autosys id for dpr: ACT#syn1#004130123DEV02 Priority: 2 Tim
e : 03/09/99 11:33:51
PID : 7668:MsgLink :9 meaningfulname :CantFindAutoSysId
 Msg: Unable to find autosys id Priority: 2 Time : 03/09/99 11:33:51
PID : 7668:MsgLink :10 meaningfulname :DpPrSchedulerDObjSmainCreateFailed
 Msg: RqFailed=CreateDpr DprID=ACT#syn1#004130123DEV02 Priority: 2 Time : 03/09/99
11:33:51
```

An "AutoSys ID" Failure means that the Job Management server could not associate the AutoSys ID with the DPR that was activated. When the Job Management server is started it reads various tables in the PDPS database that provide the linkage between processing resources and AutoSys instance. If data is missing from these tables, or was added after the Job Management server was started, then the error shown above can occur when any jobs are activated by the Planning Workbench.

The following actions should be taken when an "AutoSys ID" failure error is reported:

1. Verify that the **PlResource** table in the PDPS database has at least 1 entry for a processing string and at least one entry for an AutoSys Instance. If either of these are missing, then you need to re-do Resource Planning and add them via the Resource Editor GUI.
2. Verify that the **PlRscString** table in the PDPS database has at least 1 entry and that autosysIdKey matches the entry in the PlResource table. Again, if information is missing or wrong, you need to re-do Resource Planning.
3. Verify that the **DpPrAutosysMapList** table in the PDPS database has at least 1 entry and that **resourceString and autosysIdKey** matches the entry in the **PlRscString** table. Yet again, if information is missing or wrong, you need to re-do Resource Planning.
4. If Resource Planning has been done after the Job Management server was brought up, then bounce the server. Since the server reads this information at start up, any changes since it was brought up will NOT have taken affect.

The DPR was released but failed to be received by Job Management Server

In this case, the Planning Workbench thinks it successfully activated the DPR(s) but the Job Management Server had trouble receiving the notification.

Look for the following in the **EcDpPrJobMgmtDebug.log:**

Failed in CdsEntryRead

This indicates a problem with the communication and needs to be resolved by RTSC as a "problem with DCE". Things that you can do to confirm that this is a DCE problem:

- Run dce-verify on the machine where Job Management and Planning Workbench are executed. This should check the DCE communication service and find any errors.
- Check the Debug logs of other servers. If it is a global DCE problem, there should be errors in other server's logs such as "Invalid Bindings". When the above problem happened in the Functionality lab, all of the servers had multiple "Invalid Bindings" errors.

AutoSys is not functional

1. Set the AutoSys environment variables by sourcing something that looks like /data/autotreeb/autouser/AS1.autosys.csh,
   where AS1 is an autosys instance name.
2. type chk_auto_up and verify that you see the message: "Primary Event Processor is RUNNING on machine:
   machine-name"
3. If you don't see that a Primary Event Processor is running, check with your AutoSys administrator.

Note that if Autosys will not stay up **(Autosys administrator brings it up and it goes down right away)** the following could be occuring:

- It may be possible that too many events were queued up to AutoSys while it was down. If Autosys detects a certain number of events in a short time period, it brings itself down. The only way to handle this is to keep bringing Autosys back up. Each time it will work through a few of the events before it detects "too many" and shuts down. Eventually the events will be cleared out and Autosys will stay up.
- It may be the sql server is not up for AutoSys (this may be a different server than the one needed for the PDPS database). Look for the following error messages in the AutoSys log or when you attempt to bring up JobScape:
  Couldn't create DBPROCES
  Unable to get encoded and plaintext passwords for l0sps03_srvr:FMR

AutoSys is full

This is an unlikely problem, and would only occur under the following conditions:

- The number of job boxes in the AutoSys instance > DpPrAutoSysMaxJobs/8. Look in EcDpPrJobMgmt.CFG to get this number.
- The DPR completionState in PlDataProcessingRequest is CQ_RELEASE.

What this means is that the Job Management Server got the command from Subscription Manager to release the job, but that no more jobs can fit into AutoSys at present. Wait for a DPR to finish, so that the next waiting one can be put into AutoSys.

26.17.6.3 PDPS Troubleshooting - SDSRV Troubleshooting

How to examine the SDSRV Database

1.Log into the Science Data Server (SDSRV) machine
2.Bring up ECS Assist
3.Select "Subsystem Manager"
4.Select "ESDT Manager"
5.Select "DB Viewer"
6.Click on Login
7.A list of data types will appear. Click on the data type and information about all granules in the archive will be displayed.

PDPS Troubleshooting - Quick Tips

Job Management fails with a "JIL FAILURE" in the .ALOG file when trying to cancel a job in AutoSys

Use AutoSys Job Definition to see who owns the job.

Then select "Adv Features" to see if the user who is trying to delete the job is in the same group for "Edit Dfn" as the user who owns the job.

A quick fix is to give the job world "Edit Dfn", but generally, whoever starts the Job Management Server should be in the same group as the person using the tool to cancel the DPR.

26.18 DPREP

26.18.1 Introduction

This section contains information to schedule and run Terra DPREP.

4 Terra DPREP is made up of three PGE's that this document refers to and each are run separately. The PGE's are titled Step1 DPREP, Step 2 DPREPand Step 3 DPREP.

5 The input files come from INGEST. These files are depicted in the three step DPREP process in **Figure 26.18.1-1**

6 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. These Instrument PGE's will then process its satellite data with similar time span files created by DPREP.

7 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.

8 The SSI&T effort for DPREP PGE's is similar in effort to what would be required to register any other PGE.

26.18.2 SSI&T Activity for DPREP

The Level Zero datasets are received in 2 hour chunks. The file processes in Figure 26.18.1-1 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.

The sections that follow 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 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

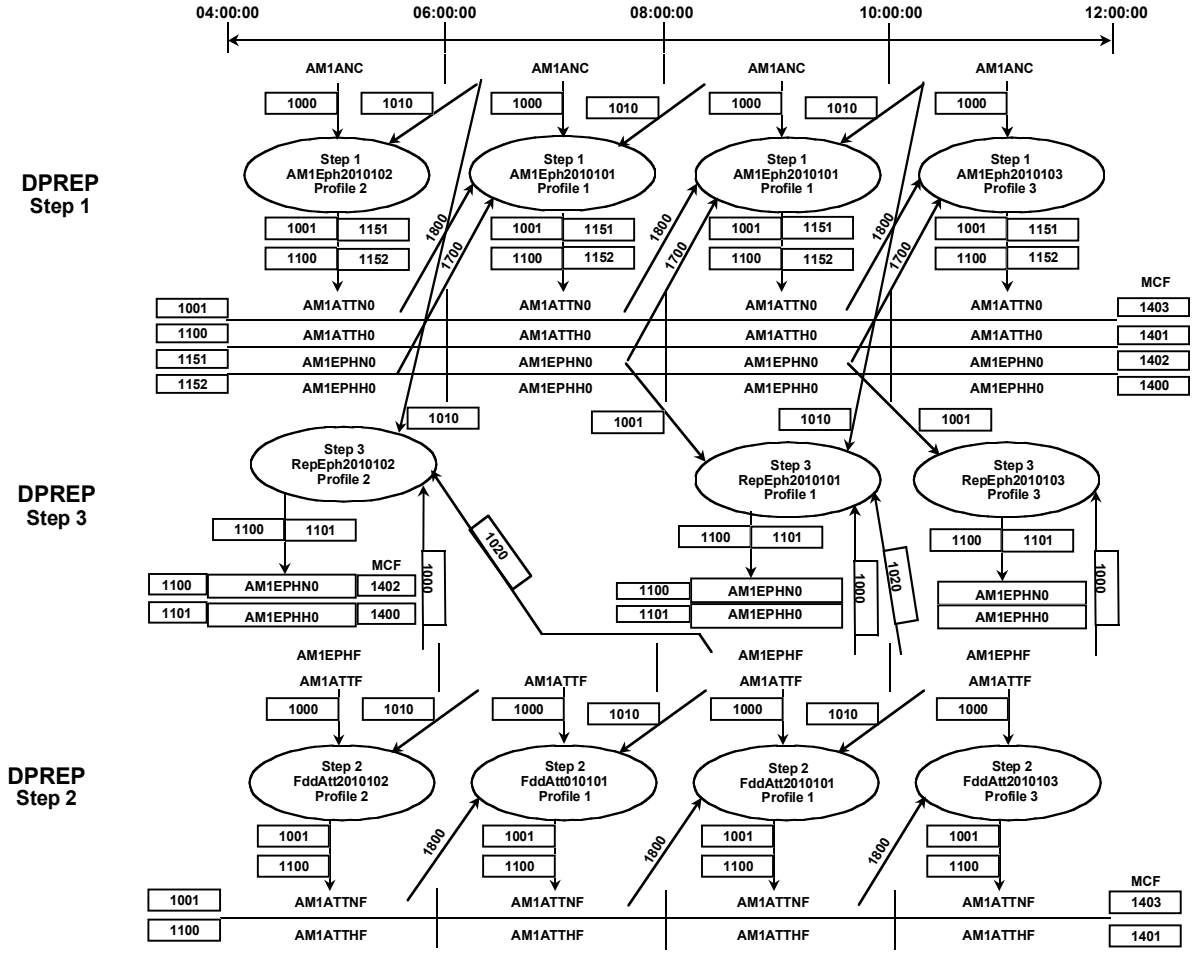


Figure 26.18.1-1. DPREP File Processes

26.18.3 DPREP Processes and Procedures

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

DPREP README and HowToRunDPREP files located at
:/usr/ecs/OPS/CUSTOM/data/DPS/

DPREP binary located: **:/usr/ecs/OPS/CUSTOM/bin/DPS/**

The following files are taken from an SSI&T SGI machine using the /data/DPS/ thread for referencing DPREP files in this section.

:

p0spg01:/usr/ecs/OPS/CUSTOM/data/DPS[45] > ls -lrt

```
total 10080
-rwxr-xr-x   1 cmops   cmops     15390 Dec 15 04:02 PGS_101
-rwxr-xr-x   1 cmops   cmops     16315 Dec 15 04:02 HowToRunDPREP
-rwxr-xr-x   1 cmops   cmops      3371 Dec 15 04:02 HowToCreateDprepTarFile
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_1.A
-rwxr-xr-x   1 cmops   cmops    450048 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_1.B
-rwxr-xr-x   1 cmops   cmops    449984 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_1.A
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_2
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_1.B
-rwxr-xr-x   1 cmops   cmops    449984 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_3
-rwxr-xr-x   1 cmops   cmops    449984 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_2
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_4
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_3
-rwxr-xr-x   1 cmops   cmops    449984 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_4
-rwxr-xr-x   1 cmops   cmops    225248 Dec 15 04:02
AM1_DEFINITIVE_ATT.fdd.Profile_3
-rwxr-xr-x   1 cmops   cmops    225248 Dec 15 04:02
AM1_DEFINITIVE_ATT.fdd.Profile_2
-rwxr-xr-x   1 cmops   cmops    225280 Dec 15 04:02
```

```
AM1_DEFINITIVE_ATT.fdd.Profile_1.B
-rwxr-xp0spg01:/usr/ecs/OPS/CUSTOM/data/DPS[45] > ls -lrt
total 10080
-rwxr-xr-x   1 cmops   cmops     15390 Dec 15 04:02 PGS_101
-rwxr-xr-x   1 cmops   cmops     16315 Dec 15 04:02 HowToRunDPREP
-rwxr-xr-x   1 cmops   cmops      3371 Dec 15 04:02 HowToCreateDprepTarFile
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_1.A
-rwxr-xr-x   1 cmops   cmops    450048 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_1.B
-rwxr-xr-x   1 cmops   cmops    449984 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_1.A
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_2
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_1.B
-rwxr-xr-x   1 cmops   cmops    449984 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_3
-rwxr-xr-x   1 cmops   cmops    449984 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_2
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_4
-rwxr-xr-x   1 cmops   cmops       384 Dec 15 04:02
EDOS_LEVEL_ZERO_00.Profile_3
-rwxr-xr-x   1 cmops   cmops    449984 Dec 15 04:02
EDOS_LEVEL_ZERO_01.Profile_4
-rwxr-xr-x   1 cmops   cmops    225248 Dec 15 04:02
AM1_DEFINITIVE_ATT.fdd.Profile_3
-rwxr-xr-x   1 cmops   cmops    225248 Dec 15 04:02
AM1_DEFINITIVE_ATT.fdd.Profile_2
-rwxr-xr-x   1 cmops   cmops    225280 Dec 15 04:02
AM1_DEFINITIVE_ATT.fdd.Profile_1.B
-rwxr-xr-x   1 cmops   cmops    225248 Dec 15 04:02
AM1_DEFINITIVE_ATT.fdd.Profile_1.A
-rwxr-xr-x   1 cmops   cmops    411600 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_2
-rwxr-xr-x   1 cmops   cmops    411600 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_1
-rwxr-xr-x   1 cmops   cmops    225248 Dec 15 04:03
AM1_DEFINITIVE_ATT.fdd.Profile_4
-rwxr-xr-x   1 cmops   cmops    411600 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_3
-rwxr-xr-x   1 cmops   cmops      1655 Dec 15 04:03 AM1ANC.Profile_4.met
-rwxr-xr-x   1 cmops   cmops      1655 Dec 15 04:03 AM1ANC.Profile_3.met
-rwxr-xr-x   1 cmops   cmops      1655 Dec 15 04:03 AM1ANC.Profile_2.met
```

```
-rwxr-xr-x  1 cmops  cmops     1655 Dec 15 04:03 AM1ANC.Profile_1.B.met
-rwxr-xr-x  1 cmops  cmops     1655 Dec 15 04:03 AM1ANC.Profile_1.A.met
-rwxr-xr-x  1 cmops  cmops     1679 Dec 15 04:03
AM1_DEFINITIVE_ATT.fdd.Profile_4.met
r-x  1 cmops  cmops   225248 Dec 15 04:02
AM1_DEFINITIVE_ATT.fdd.Profile_1.A
-rwxr-xr-x  1 cmops  cmops   411600 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_2
-rwxr-xr-x  1 cmops  cmops   411600 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_1
-rwxr-xr-x  1 cmops  cmops   225248 Dec 15 04:03
AM1_DEFINITIVE_ATT.fdd.Profile_4
-rwxr-xr-x  1 cmops  cmops   411600 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_3
-rwxr-xr-x  1 cmops  cmops     1655 Dec 15 04:03 AM1ANC.Profile_4.met
-rwxr-xr-x  1 cmops  cmops     1655 Dec 15 04:03 AM1ANC.Profile_3.met
-rwxr-xr-x  1 cmops  cmops     1655 Dec 15 04:03 AM1ANC.Profile_2.met
-rwxr-xr-x  1 cmops  cmops     1655 Dec 15 04:03 AM1ANC.Profile_1.B.met
-rwxr-xr-x  1 cmops  cmops     1655 Dec 15 04:03 AM1ANC.Profile_1.A.met
-rwxr-xr-x  1 cmops  cmops     1679 Dec 15 04:03
AM1_DEFINITIVE_ATT.fdd.Profile_4.met
-rwxr-xr-x  1 cmops  cmops     1679 Dec 15 04:03
AM1_DEFINITIVE_ATT.fdd.Profile_3.met
-rwxr-xr-x  1 cmops  cmops     1679 Dec 15 04:03
AM1_DEFINITIVE_ATT.fdd.Profile_2.met
-rwxr-xr-x  1 cmops  cmops     1679 Dec 15 04:03
AM1_DEFINITIVE_ATT.fdd.Profile_1.B.met
-rwxr-xr-x  1 cmops  cmops     1679 Dec 15 04:03
AM1_DEFINITIVE_ATT.fdd.Profile_1.A.met
-rwxr-xr-x  1 cmops  cmops     5909 Dec 15 04:03 ReplaceEphemeris.tar.met
-rwxr-xr-x  1 cmops  cmops     5909 Dec 15 04:03 FDDAttitude.tar.met
-rwxr-xr-x  1 cmops  cmops    29206 Dec 15 04:03
EcDpPrAm1EdosEphAttDPREP_PGE.Steps1ab.PCF.Profile_3
-rwxr-xr-x  1 cmops  cmops    29299 Dec 15 04:03
EcDpPrAm1EdosEphAttDPREP_PGE.Steps1ab.PCF.Profile_2
-rwxr-xr-x  1 cmops  cmops    29818 Dec 15 04:03
EcDpPrAm1EdosEphAttDPREP_PGE.Steps1ab.PCF.Profile_1.B
-rwxr-xr-x  1 cmops  cmops    29818 Dec 15 04:03
EcDpPrAm1EdosEphAttDPREP_PGE.Steps1ab.PCF.Profile_1.A
-rwxr-xr-x  1 cmops  cmops     1663 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_3.met
-rwxr-xr-x  1 cmops  cmops     1663 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_2.met
-rwxr-xr-x  1 cmops  cmops     1663 Dec 15 04:03
AM1_REPAIR_EPH.fdd.Profile_1.met
```

-rwxr-xr-x 1 cmops cmops 5909 Dec 15 04:03
AM1_Ancillary_DPREP.tar.met
-rwxr-xr-x 1 cmops cmops 27664 Dec 15 04:03
EcDpPrAm1FddEphemerisDPREP_PGE.Step3.PCF.Profile_2
-rwxr-xr-x 1 cmops cmops 27944 Dec 15 04:03
EcDpPrAm1FddEphemerisDPREP_PGE.Step3.PCF.Profile_1
-rwxr-xr-x 1 cmops cmops 26902 Dec 15 04:03
EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.Profile_4
-rwxr-xr-x 1 cmops cmops 27257 Dec 15 04:03
EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.Profile_3
-rwxr-xr-x 1 cmops cmops 27184 Dec 15 04:03
EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.Profile_2
-rwxr-xr-x 1 cmops cmops 27543 Dec 15 04:03
EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.Profile_1.B
-rwxr-xr-x 1 cmops cmops 27543 Dec 15 04:03
EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.Profile_1.A
-rwxr-xr-x 1 cmops cmops 28695 Dec 15 04:03
EcDpPrAm1EdosEphAttDPREP_PGE.Steps1ab.PCF.Profile_4
-rwxr-xr-x 1 cmops cmops 26584 Dec 15 04:03
EcDpPrAm1FddEphemerisDPREP_PGE.Step3.PCF.Profile_3
-rwxr-xr-x 1 cmops cmops 5617 Dec 15 04:03 DPREP_README
-rw-rw-r-- 1 cmshared cmshared 433 Dec 15 15:35 ODL
p0spg01:/usr/ecs/OPS/CUSTOM/data/DPS[46] >
data3/ecs/OPS/CUSTOM/data/DPS

26.18.3.1 DPREP consists of three PGE's each run separately.

1 The first step is a **ksh** script called **EcDpPrAm1EdosEphAttDPREP_PGE**, which serves as a driver for three executables:
   EcDpPrAm1EdosAncillary
   EcDpPrAm1EdosEphemerisRepair
   EcDpPrAm1ToolkitHdf
2 The second step is **EcDpPrAm1FddAttitudeDPREP_PGE.**
3 The third step is **EcDpPrAm1FddEphemerisDPREP_PGE**.

STEP ONE

EcDpPrAm1EdosAncillary reads in AM-1 L0 (EDOS) Ancillary Dataset (logical id 1000, ESDT AM1ANC). It also reads another set of AM-1 L0 (EDOS) Ancillary Dataset (logical id 1010, ESDT AM1ANC). The second set of L0 data is required to insure that incomplete orbits in the first data set get complete orbit metadata records. The only data that will be extracted from the second data set is the descending node time and longitude.

EcDpPrAm1EdosAncillary also reads in ephemeris and attitude data (toolkit native format) under logical ids 1700 (ESDT AM1EPHN0) and 1800 (ESDT AM1ATTN0). These would be the last ephemeris/attitude data sets generated from a previous run of this PGE.

EcDpPrAm1EdosEphemerisRepair

If EcDpPrAm1EdosAncillary signals a short gap was detected then EcDpPrAm1EdosEphemerisRepair reads the scratchfile created by EcDpPrAm1EdosAncillary

and performs the gap fill and writes a gap filled native format ephemeris file. If no short gap is signaled then the scratch file is simply renamed to the native format ephemeris file.

EcDpPrAm1ToolkitHdf

This process takes the native format ephemeris file and produces a corresponding HDF file and a metadata file.

This PGE produces Toolkit and HDF format attitude and ephemeris data sets. The attitude data sets are produced using logical ids 1001 (ESDT AM1ATTN0) and 1100 (ESDT AM1ATTH0). The ephemeris data sets are produced using logical ids 1101 (ESDT AM1EPHN0) and 1102 (ESDT AM1EPHH0). The corresponding MCFs are accessed using logical ids 1400 (ESDT AM1EPHH0), 1401 (ESDT AM1ATTH0), 1402 (ESDT AM1EPHN0) and 1403 (ESDT AM1ATTN0).

The PGE produces an ASCII report file under logical id 2000

EcDpPrAm1EdosEphAttDPREP_PGE. Steps1ab.PCF.curr is a sample pcf used by this PGE. This pcf has directory locations and file names which may not be valid in all environments. Be careful before directly using this pcf. A sample set of input files for this PGE are

1000 EDOS_LEVEL_ZERO_00.curr
1000 EDOS_LEVEL_ZERO_01.curr
1010 EDOS_LEVEL_ZERO_00.next
1010 EDOS_LEVEL_ZERO_01.next

and a copy of each of these files is provided. The files in this group do contain small data gaps which will cause EcDpPrAm1EdosEphemerisRepair to be run.

The remaining pcfs for Step 1 DPREP:

EcDpPrAm1EdosEphAttDPREP_PGE.Steps1ab.PCF.next
EcDpPrAm1EdosEphAttDPREP.Steps1ab.PCF.next_1

will move through the sequence of the Level Zero files provided.

example; EDOS_LEVEL_ZERO_00.next is treated as current and EDOS_LEVEL_ZERO_00.next_1 is treated as next etc...

These pcfs can be utilized at the testers discretion and are not necessary for the running of steps 2 and 3. The input files indicated in EcDpPrAm1EdosEphAttDPREP.Steps1ab.PCF.next_1 contain no gaps so testers shouldn't be concerned that EcDpPrAm1EdosEphemerisRepair is not invoked when using this pcf.

Note:

The Level Zero datasets are received in 2 hour chunks but processing can't be performed on the most recently available 2 hour chunk. Step 1 processing needs to look forward in the time stream in order to complete orbit metadata processing.

STEP TWO

EcDpPrAm1FddAttitudeDPREP_PGE reads in the current FDD Attitude Dataset under logical ID 1000 and the next FDD Attitude Dataset under logical ID 1010. It also reads in the attitude data set it produced with it's last run under logical ID 1502. The output of this process is a native format attitude file (logical ID 1001) and an HDF format attitude file (logical ID 1100). A .met file is also produced for each.

EcDpPrAm1FddAttitudeDPREP_PGE.Step2.PCF.curr is a sample pcf used by this PGE. This pcf has directory locations and file names which may not be valid in all environments. Be careful before directly using this pcf. A sample set of input files for this PGE are

1000 AM1_DEFINITIVE_ATT.fdd.curr
1010 AM1_DEFINITIVE_ATT.fdd.next
(IMPORTANT: These files contain incorrect data and were delivered only as a placeholder. EcDpPrAm1FddAttitudeDPREP_PGE will not run with these files as input. When valid FDD Definitive Attitude files become available they should be moved to these filenames and EcDpPrAm1FddAttitudeDPREP_PGE should run successfully.)
and a copy of each of these files is provided.

Note: Step 2 processing must follow 2 hours behind step 1 processing.

STEP THREE

If step1 finds too many missing data points in the ephemeris data it signals that a definitve ephemeris file is needed from FDD which EcDpPrAm1FddEphemerisDPREP_PGE will use to replace the ephemeris dataset that was generated from the Level Zero data.
EcDpPrAm1FddEphemerisDPREP_PGE reads in the definitive Ephemeris dataset received from FDD (logical ID 1000) and the EOS_AM1 Ephemeris data (logical id 1001) in toolkit native format.
It produces Replacement ephemeris datasets (logical id 1101, ESDT AM1EPHH0) and (logical id 1100, ESDT AM1EPHN0). The corresponding MCFs are accessed using logical ids 1400 (ESDT AM1EPHH0) & 1402 (ESDT AM1EPHN0)

EcDpPrAm1FddEphemerisDPREP_PGE.Step3.PCF.curr is a sample pcf used by this PGE. This pcf has directory locations and file names which may not be valid in all environments. Be careful before directly using this pcf. A sample FDD input file for this PGE is
1000 AM1_REPAIR_EPH.fdd
and a copy of this file is provided.
Toolkit Initialization Settings
SGI Application Binary Interface New 32
Disk Space Used for PGE Run 50.000 MB
Shared Memory ON
Use Test Files if SM Fails ON
Use Log Files ON
Continue if Logging Fails OFF

26.18.4 Creating DPREP Tar Files

If tar files are not available for registering the three AM-1 DPREP PGEs,

Follow the instructions found in the HowToCreateDprepTarFiles file to generate the necessary tar files.

26.18.4.1 HowToCreateDprepTarFiles

p0spg01:/usr/ecs/OPS/CUSTOM/data/DPS[46] > more HowToCreateDprepTarFile
This file provides information to create the DPREP pge tar files

Objective : To create AM1_Ancillary_DPREP.tar & FddAttitude.tar
and using SSIT Manager archive the PGE tar files

Files needed :

PGS_101
EcDpPrAm1EdosEphAttDPREP_PGE
EcDpPrAm1EdosAncillary
EcDpPrAm1EdosEphemerisRepair
EcDpPrAm1ToolkitToHdf
EcDpPrAm1FddAttitudeDPREP_PGE
EcDpPrAm1FddEphemerisDPREP_PGE

Steps to create AM1_Ancillary_DPREP.tar

1. Login to the science processing machine **(SGI)**
2. Check to make sure that the executables
EcDpPrAm1EdosEphAttDPREP_PGE
EcDpPrAm1EdosAncillary
EcDpPrAm1EdosEphemerisRepair
EcDpPrAm1ToolkitToHdf
are delivered and reside in **$ECS_HOME/<MODE>/CUSTOM/bin/DPS**
3. Check to make sure that the data file **PGS_101** is delivered and resides in **$ECS_HOME/<MODE>/CUSTOM/data/DPS**

4. cd to $ECS_HOME/<MODE>/CUSTOM/data/DPS
5. /bin/tar cvf AM1_Ancillary_DPREP.tar PGS_101 -C $ECS_HOME/<MODE>/CUSTOM/bin/DPS EcDpPrAm1EdosAncillary EcDpPrAm1EdosEphAttDPREP_PGE EcDpPrAm1EdosEphemerisRepair EcDpPrAm1ToolkitToHdf
Step 5 will create the **AM1_Ancillary_DPREP.tar** file in **$ECS_HOME/<MODE>/CUSTOM/data/DPS**
This file then needs to be ftped to the SSIT machine from where it will be archived
The corresponding met file for this tar file is **AM1_Ancillary_DPREP.tar.met**

and is delivered to **$ECS_HOME/<MODE>/CUSTOM/data/DPS**
This file then needs to be ftped to the SSIT machine from where it (and the tar file itself) will be archived

Steps to create FddAttitude.tar

1. Login to the science processing machine **(SGI)**
2. Check to make sure that the executable **EcDpPrAm1FddAttitudeDPREP_PGE** is delivered and resides in **$ECS_HOME/<MODE>/CUSTOM/bin/DPS**
3. Check to make sure that the data file **PGS_101** is delivered and resides in **$ECS_HOME/<MODE>/CUSTOM/data/DPS**

4. cd to $ECS_HOME/<MODE>/CUSTOM/data/DPS

5. /bin/tar cvf FDDAttitude.tar PGS_101 -C $ECS_HOME/<MODE>/CUSTOM/bin/DPS EcDpPrAm1FddAttitudeDPREP_PGE

Step 5 will create the **FDDAttitude.tar file** in **$ECS_HOME/<MODE>/CUSTOM/data/DPS**
This file then needs to be ftped to the SSIT machine from where it will be archived
The corresponding met file for this tar file is **FDDAttitude.tar.met** and is delivered to **$ECS_HOME/<MODE>/CUSTOM/data/DPS**
This file then needs to be ftped to the SSIT machine from where it (and the corresponding tar file) will be archived**.**

Steps to create ReplaceEphemeris.tar

1. Login to the science processing machine (SGI)
2. Check to make sure that the executable **EcDpPrAm1FddEphemerisDPREP_PGE** is delivered and resides in **$ECS_HOME/<MODE>/CUSTOM/bin/DPS**
3. Check to make sure that the data file **PGS_101** is delivered and resides in **$ECS_HOME/<MODE>/CUSTOM/data/DPS**
4. cd to $ECS_HOME/<MODE>/CUSTOM/data/DPS
5. /bin/tar cvf ReplaceEphemeris.tar PGS_101 -C $ECS_HOME/<MODE>/CUSTOM/bin/DPS EcDpPrAm1FddEphemerisDPREP_PGE
Step 5 will create the ReplaceEphemeris.tar file in **$ECS_HOME/<MODE>/CUSTOM/data/DPS**
This file then needs to be ftped to the SSIT machine from where it will be archived
The corresponding met file for this tar file is ReplaceEphemeris.tar.met and is delivered to **$ECS_HOME/<MODE>/CUSTOM/data/DPS**
This file then needs to be ftped to the SSIT machine from where it

(and the corresponding tar file) will be archived.

26.18.4.2 HowToRunDPREP

p0spg01:/usr/ecs/OPS/CUSTOM/data/DPS[48] > more HowToRunDPREP

This document contains information on how to schedule and run **AM-1 DPREP**
AM-1 DPREP is made up of **3 PGEs** that this document refers to as **Step 1 DPREP, Step 2 DPREP**, and **Step 3 DPREP:**

Step 1 DPREP is the **EDOS-**supplied ephemeris and attitude preprocessor**.**
Step 2 DPREP is the **FDD-**supplied attitude preprocessor**.**
Step 3 DPREP is the **FDD-**supplied ephemeris preprocessor**.**

Operationally, Steps 1 and 2 are scheduled daily and run independently of one another. Step 3 is scheduled and run on an as-needed basis.
In order to run, **DPREP** processing expects data to be available not only from the current segment, but also from the preceding and following segments as well. Data from the preceding and following segments are used to consistency check ephemeris and attitude data streams when the data streams bridge segment boundaries. The availability of data from the preceding and following segments is not guaranteed, however. Four data processing profiles have been developed to handle the various permutations of preceding and following data segments that could be available to **DPREP:**

Profile 1 expects data from the preceding, current, and following segments.
Profile 2 expects data from the current and following segments only.
Profile 3 expects data from the preceding and current segments only.
Profile 4 expects data from the current segment only.

These profiles are recognized by all **DPREP Steps.** The **ESDT** that is provided to **DPREP** from the preceding, current, and following data segments depends on the Step being run, however.

Profile 2 (no preceding data, but following data is available) initializes **DPREP's** processing of a given Step's ephemeris and/or attitude data stream. **Once Profile 2** has been run on a data segment, **Profile 1** (preceding and following data available) assumes processing responsibility on all data segments thereafter until data dropout or mission end is encountered. **Profile 3** (preceding data available, but no following data) then processes data segment that immediately precedes data dropout and, therefore, terminates processing on a given Step's ephemeris and/or attitude data stream**.**

In the big picture then, DPREP processing requires a single **Profile 2** to run on the first data segment, is followed by an indeterminant number of **Profile 1** processes, and is terminated by a single **Profile 3** process**.**

Profile 4 processes isolated data segments and is not likely to be scheduled operationally.

More information on the **AM-1 DPREP** can be found in document **"TERRA Spacecraft Ephemeris** and **Attitude Data Preprocessing"** (document number **184-TP-001-001).**

Note: Each PGE will use both the **Science Metadata update tool** and the **Operational Metadata Update tool,** which are located in the **SSIT Manager,** to incorporate the specific parameters listed in each of the direction sections below.

26.18.4.3 Registering DPREP PGEs

So that the following tests can be conducted, all four Profiles must be registered for all three DPREP Steps. Each DPREP PGE is registered once.

Step 1 DPREP Test Instructions

For Step 1, **current** and **following** granules are EDOS-supplied L0 Ancillary (APID 4, ShortName AM1ANC), the preceding granule is the output Toolkit format ephemeris and attitude products produced by the preceding run of Step 1 (ShortNames AM1EPHN0, AM1ATTN0). DPREP expects both the L0 Ancillary header granule and the data granule (hence, 2 granules) for the current and following L0 Ancillary. Step 1 requires 2 preceding data sets, one ephemeris product and one attitude product from the preceding Step 1 run (two granules, one for ephemeris and one for attitude).

Step 1 DPREP Specifications

Tar File Name : $ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_Ancillary_DPREP.tar
Contents : EcDpPrAm1EdosEphAttDPREP_PGE
EcDpPrAm1EdosAncillary
EcDpPrAm1EdosEphemerisRepair
EcDpPrAm1ToolkitToHdf
PGS_101
PGE Met File :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_Ancillary_DPREP.tar.met

Top Level Exe : EcDpPrAm1EdosEphAttDPREP_PGE

PGE Name : AM1Eph
Version Number : 20202

Profile 1 ODL : PGE_AM1Eph#2020101.odl
Profile 2 ODL : PGE_AM1Eph#2020102.odl
Profile 3 ODL : PGE_AM1Eph#2020103.odl

Profile 4 ODL : PGE_AM1Eph#2020104.odl

Test Scenario
The Step 1 DPREP test is divided into two parts. The first part exercises Profiles 1, 2, and 3 over four consecutive data segments. The first data segment is scheduled for Profile 2 processing, the next two segments are scheduled for Profile 1 processing, and the last scheduled for Profile 3 processing. The data segments to be processed follow.

Segment 1: 1997 July 31 04:00:00 to 1997 July 31 06:00:00 (Profile 2)
Segment 2: 1997 July 31 06:00:00 to 1997 July 31 08:00:00 (Profile 1)
Segment 3: 1997 July 31 08:00:00 to 1997 July 31 10:00:00 (Profile 1)
Segment 4: 1997 July 31 10:00:00 to 1997 July 31 12:00:00 (Profile 3)

The second Step 1 DPREP test exercises Profile 4 on an isolated data segment.

Segment 5: 1997 July 31 18:00:00 to 1997 July 31 20:00:00 (Profile 4)

There is one dynamic granule (2 files for each granule) for each of the data segments described above. Insert all five dynamic granules into the archive using the SSIT Manager's insert dynamic data tool.

Segment 1:

ESDT Short Name : AM1ANC
File Names : EDOS_LEVEL_ZERO_00.Profile_2
EDOS_LEVEL_ZERO_01.Profile_2
Met file : AM1ANC.Profile_2.met
Time Range : 1997 July 31 04:00:00 to 1997 July 31 06:00:00

Segment 2:

ESDT Short Name : AM1ANC
File Names : EDOS_LEVEL_ZERO_00.Profile_1.A
EDOS_LEVEL_ZERO_01.Profile_1.A
Met file : AM1ANC.Profile_1.A.met
Time Range : 1997 July 31 06:00:00 to 1997 July 31 08:00:00

Segment 3:
ESDT Short Name : AM1ANC
File Names : EDOS_LEVEL_ZERO_00.Profile_1.B
EDOS_LEVEL_ZERO_01.Profile_1.B
Met file : AM1ANC.Profile_1.B.met

Time Range : 1997 July 31 08:00:00 to 1997 July 31 10:00:00

Segment 4:
ESDT Short Name : AM1ANC
File Names : EDOS_LEVEL_ZERO_00.Profile_3
EDOS_LEVEL_ZERO_01.Profile_3
Met file : AM1ANC.Profile_3.met
Time Range : 1997 July 31 10:00:00 to 1997 July 31 12:00:00

Segment 5:
ESDT Short Name : AM1ANC
File Names : EDOS_LEVEL_ZERO_00.Profile_4
EDOS_LEVEL_ZERO_01.Profile_4
Met file : AM1ANC.Profile_4.met
Time Range : 1997 July 31 18:00:00 to 1997 July 31 20:00:00

Create three production requests for the following time ranges. The number of DPRs that should be created are given below.

Request	Timespan	Profile	DPRs
1	1997 July 31 04:00:00 to 06:00:00	2	1
2	1997 July 31 06:00:00 to 10:00:00	1	2
3	1997 July 31 10:00:00 to 12:00:00	3	1

Schedule all three production requests to run as a single production plan. This completes the first part of the Step 1 DPREP test.

Next, create one production request that for the time range that follow.

Request	Timespan	Profile	DPRs
1	1997 July 31 18:00:00 to 20:00:00	4	1

Schedule this a seperate production plan. This completes the second part of the Step 1 DPREP test.

Results
Find output granules for each of the timespans described above for ESDTs AM1EPHN0, AM1EPHH0, AM1ATTN0, and AM1ATTH0. ESDT AM1EPHN0 is the Toolkit format ephemeris granule, while ESDT AM1EPHH0 is the HDF format ephemeris granule. ESDT AM1ATTN0 is the Toolkit format attitude granule, while ESDT AM1ATTH0 is the HDF format attitude granule.

Step 2 DPREP Test Instructions

For Step 2, current and following granules are FDD-supplied attitude (ShortName

AM1ATTF), the preceding granule is the output Toolkit format attitude product produced by the preceding run of Step 2 (ShortName AM1ATTNF).

Step 2 DPREP Specifications

Tar File Name : $ECS_HOME/<MODE>/CUSTOM/data/DPS/FDDAttitude.tar
Contents : EcDpPrAm1FddAttitudeDPREP_PGE
PGS_101
PGE Met File : $ECS_HOME/<MODE>/CUSTOM/data/DPS/FDDAttitude.tar.met

Top Level Exe : EcDpPrAm1FddAttitudeDPREP_PGE

PGE Name : FddAtt
Version Number : 20201

Profile 1 ODL : PGE_FddAtt#2020101.odl
Profile 2 ODL : PGE_FddAtt#2020102.odl
Profile 3 ODL : PGE_FddAtt#2020103.odl
Profile 4 ODL : PGE_FddAtt#2020104.odl

Test Scenario

The Step 2 DPREP test, like the Step 1 test scenario, is divided into two parts. The first part exercises Profiles 1, 2, and 3 over four consecutive data segments. The first data segment is scheduled for Profile 2 processing, the next two segments are scheduled for Profile 1 processing, and the last scheduled for Profile 3 processing. The data segments to be processed follow.

Segment 1: 1997 July 31 04:00:00 to 1997 July 31 06:00:00 (Profile 2)
Segment 2: 1997 July 31 06:00:00 to 1997 July 31 08:00:00 (Profile 1)
Segment 3: 1997 July 31 08:00:00 to 1997 July 31 10:00:00 (Profile 1)
Segment 4: 1997 July 31 10:00:00 to 1997 July 31 12:00:00 (Profile 3)

The second Step 2 DPREP test exercises Profile 4 on an isolated data segment.

Segment 5: 1997 July 31 18:00:00 to 1997 July 31 20:00:00 (Profile 4)

There is one granule for each of the data segments described above. Insert all five granules into the archive using the SSIT Manager's insert dynamic data tool.

Segment 1:

ESDT Short Name : AM1ATTF
File Names : AM1_DEFINITIVE_ATT.fdd.Profile_2
Met file : AM1_DEFINITIVE_ATT.fdd.Profile_2.met
Time Range : 1997 July 31 04:00:00 to 1997 July 31 06:00:00

Segment 2:

ESDT Short Name : AM1ATTF
File Names : AM1_DEFINITIVE_ATT.fdd.Profile_1.A
Met file : AM1_DEFINITIVE_ATT.fdd.Profile_1.A.met
Time Range : 1997 July 31 06:00:00 to 1997 July 31 08:00:00

Segment 3:

ESDT Short Name : AM1ATTF
File Names : AM1_DEFINITIVE_ATT.fdd.Profile_1.B
Met file : AM1_DEFINITIVE_ATT.fdd.Profile_1.B.met
Time Range : 1997 July 31 08:00:00 to 1997 July 31 10:00:00

Segment 4:
ESDT Short Name : AM1ATTF
File Names : AM1_DEFINITIVE_ATT.fdd.Profile_3
Met file : AM1_DEFINITIVE_ATT.fdd.Profile_3.met
Time Range : 1997 July 31 10:00:00 to 1997 July 31 12:00:00

Segment 5:
ESDT Short Name : AM1ATTF
File Names : AM1_DEFINITIVE_ATT.fdd.Profile_4
Met file : AM1_DEFINITIVE_ATT.fdd.Profile_4.met
Time Range : 1997 July 31 18:00:00 to 1997 July 31 20:00:00

Create three production requests for the following time ranges. The number of DPRs that should be created are given below.

Request	Timespan	Profile	DPRs
1	1997 July 31 04:00:00 to 06:00:00	2	1
2	1997 July 31 06:00:00 to 10:00:00	1	2
3	1997 July 31 10:00:00 to 12:00:00	3	1

Schedule all three production requests to run as a single production plan. This completes the first part of the Step 2 DPREP test.

Next, create one production request that for the time range that follow.

Request	Timespan	Profile	DPRs
1	1997 July 31 18:00:00 to 20:00:00	4	1

Schedule this a seperate production plan. This completes the second part of the Step 2 DPREP test.

These plans can be scheduled and run completely independently of the Step 1 DPREP plans described earlier.

Results
Find output granules for each of the timespans described above for ESDTs AM1ATTNF and AM1ATTHF. ESDT AM1ATTNF is the Toolkit format attitude granule, while ESDT AM1ATTHF is the HDF format attitude granule.

Step 3 DPREP Test Instructions

For Step 3, the current granule is FDD-supplied ephemeris (ShortName AM1EPHF), the preceding granule is the output Toolkit format ephemeris product produced by preceding run of Step 1, and the following granule is EDOS-supplied L0 Ancillary (ShortName AM1ANC).

Step 3 DPREP Specifications
Tar File Name : $ECS_HOME/<MODE>/CUSTOM/data/DPS/ReplaceEphemeris.tar
Contents : EcDpPrAm1FddAttitudeDPREP_PGE
PGS_101
PGE Met File : $ECS_HOME/<MODE>/CUSTOM/data/DPS/ReplaceEphemeris.tar.met

Top Level Exe : EcDpPrAm1FddEphemerisDPREP_PGE

PGE Name : RepEph
Version Number : 20201

Profile 1 ODL : PGE_RepEph#2020101.odl
Profile 2 ODL : PGE_RepEph#2020102.odl
Profile 3 ODL : PGE_RepEph#2020103.odl
Profile 4 ODL : PGE_RepEph#2020104.odl

Test Scenario

Operationally, Step 3 AM-1 DPREP will most likely be scheduled to run on single data segments. Hence, the Step 3 test scenario consists of scheduling and running four plans that exercise the four Profiles individually on a single

data segments. The data segments to be processed follow.

Segment 1: 1997 July 31 04:00:00 to 1997 July 31 06:00:00 (Profile 2)
Segment 2: 1997 July 31 08:00:00 to 1997 July 31 10:00:00 (Profile 1)
Segment 3: 1997 July 31 10:00:00 to 1997 July 31 12:00:00 (Profile 3)
Segment 4: 1997 July 31 18:00:00 to 1997 July 31 20:00:00 (Profile 4)

There is one granule for each of the data segments described above. Insert all four granules into the archive using the SSIT Manager's insert dynamic data tool.

Segment 1:

ESDT Short Name : AM1EPHF
File Names : AM1_REPAIR_EPH.fdd.Profile_2
Met file : AM1_REPAIR_EPH.fdd.Profile_2.met
Time Range : 1997 July 31 04:00:00 to 1997 July 31 06:00:00

Segment 2:
ESDT Short Name : AM1EPHF
File Names : AM1_REPAIR_EPH.fdd.Profile_1
Met file : AM1_REPAIR_EPH.fdd.Profile_1.met
Time Range : 1997 July 31 08:00:00 to 1997 July 31 10:00:00

Segment 3:
ESDT Short Name : AM1EPHF
File Names : AM1_REPAIR_EPH.fdd.Profile_3
Met file : AM1_REPAIR_EPH.fdd.Profile_3.met
Time Range : 1997 July 31 10:00:00 to 1997 July 31 12:00:00

Segment 4:
ESDT Short Name : AM1EPHF
File Names : AM1_REPAIR_EPH.fdd.Profile_4
Met file : AM1_REPAIR_EPH.fdd.Profile_4.met
Time Range : 1997 July 31 18:00:00 to 1997 July 31 20:00:00

Create four individual production plans consisting of a single DPR for each of the time ranges that follow. Schedule and allow a given plan to complete before scheduling the next.

Request	Timespan	Profile	DPRs
1	1997 July 31 04:00:00 to 06:00:00	2	1
2	1997 July 31 06:00:00 to 10:00:00	1	1
3	1997 July 31 10:00:00 to 12:00:00	3	1
4	1997 July 31 18:00:00 to 20:00:00	4	1

These plans can be scheduled and run only after the Step 1 AM-1 DPREP test has been successfully completed. The Step 1 DPREP test scenario must be completed before CONTINUING with the Step 3 test scenario. Continuation of the Step 3 test scenario requires the archive environment that was produced from the successful completion of the Step 1 DPREP test scenario.

Results
Find output granules for each of the timespans described above for ESDTs AM1EPHN0 and AM1EPHH0. ESDT AM1EPHN0 is the Toolkit format ephemeris granule, while ESDT AM1EPHH0 is the HDF format ephemeris granule. Output from Step 3 processing is placed in the same ESDTs as the Step 1 products.

26.18.5 Results of DPREP processing from earlier work

26.18.5.1 Step1 DPREP directions taken from earlier work:

There are 2 profiles for DPREP. Profile 1 takes in previous DPREP output and is expected to be run most of the time at the DAACs. Profile 2 takes in only the AM1 Ancillary data and will be run only for the first run of DPREP (because there is no previous output). Both profiles should be registered and executed.
PGENAME : AM1Eph
PGEVERSION : 2.0
PROFILE : 1
TopLevelShellName : EcDpPrAm1EdosEphAttDPREP_PGE
PGENAME : AM1Eph
PGEVERSION : 2.0
PROFILE : 2
TopLevelShellName : EcDpPrAm1EdosEphAttDPREP_PGE (same executable, only insert it once)
PGE Tar file location :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_Ancillary_DPREP.tar
PGE Met file location :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_Ancillary_DPREP.tar.met
Science Software Release 4
The PGE tar file contains the executables: EcDpPrAm1EdosEphAttDPREP_PGE, EcDpPrAm1EdosAncillary, EcDpPrAm1EdosEphemerisRepair, EcDpPrAm1ToolkitHdf.and the toolkit message file PGS_101
This PGE has no static data. This means Insert Static from SSIT can be skipped for this PGE.

However, there are 4 dynamic granules (2 files each) that this PGE needs as input. These 4 granules can be inserted from SSIT **Insert Test Dynamic Tool**

Granule 1:

ESDT Short Name : AM1ANC
Version : 001
Multi File granule : Y
Directory : $ECS_HOME/<MODE>/CUSTOM/data/DPS
File Names : EDOS_LEVEL_ZERO_00.curr, EDOS_LEVEL_ZERO_01.curr
Met file : AM1ANC.curr.met

Granule 2:
ESDT Short Name : AM1ANC
Version : 001
Multi File granule : Y
Directory : $ECS_HOME/<MODE>/CUSTOM/data/DPS
File Names : EDOS_LEVEL_ZERO_00.next, EDOS_LEVEL_ZERO_01.next
Met file : AM1ANC.next.met

Granule 3:
ESDT Short Name : AM1ANC
Version : 001
Multi File granule : Y
Directory : $ECS_HOME/<MODE>/CUSTOM/data/DPS
File Names : EDOS_LEVEL_ZERO_00.next_1, EDOS_LEVEL_ZERO_01.next_1
Met file : AM1ANC.next_1.met

Granule 4:
ESDT Short Name : AM1ANC
Version : 001
Multi File granule : Y
Directory : $ECS_HOME/<MODE>/CUSTOM/data/DPS
File Names : EDOS_LEVEL_ZERO_00.next_2, EDOS_LEVEL_ZERO_01.next_2
Met file : AM1ANC.next_2.met
3 PRs need to be created, two for Profile 1 and one for Profile 2. Create the PR for Profile 2 first (it runs on earlier data) and then create the PR for Profile 1.

The time frame for Production request editor for Profile 2 is 1998 June 30 00:00:00 to 1998 June 30 02:00:00

The time frame for Production request editor for Profile 1 is 1998 June 30 02:00:00 to 1998 June 30 04:00:00

The time frame for Production request editor for the second Profile 1 is 1998 June 30 04:00:00 to 1998 June 30 06:00:00

26.18.5.2 Step2 DPREP directions:

****

Step 2 DPREP can run ONLY after all three PRs for Step 1 DPREP have completed

****

There are 2 profiles for Step 2 DPREP. Profile 1 takes in previous Step 2 DPREP output and is expected to be run most of the time at the DAACs. Profile 2 takes in only the Fdd Att data and

will be run only for the first run of DPREP (because there is no previous output). Both profiles should be registered and executed.

PGENAME : FddAtt
PGEVERSION : 1.0
PROFILE : 1
TopLevelShellName : EcDpPrAm1FddAttitudeDPREP_PGE

PGENAME : FddAtt
PGEVERSION : 1.0
PROFILE : 2
TopLevelShellName : EcDpPrAm1FddAttitudeDPREP_PGE (same executable, only insert it once)

PGE Tar file location : $ECS_HOME/<MODE>/CUSTOM/data/DPS/FDDAttitude.tar
PGE Met file location : $ECS_HOME/<MODE>/CUSTOM/data/DPS/FDDAttitude.tar.met
Science Software Version: 1.0

The PGE tar file contains the executable EcDpPrAm1FddAttitudeDPREP_PGE and the toolkit message file PGS_101
This PGE has no static data. This means Insert Static from SSIT can be skipped for this PGE. However, there are 3 dynamic granules that this PGE needs as input. These 3 granules can be inserted from SSIT insert test dynamic tool

Granule 1:
ESDT Short Name : AM1ATTF
Version : 001
Multi File granule : N
File Name :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.curr
Met file :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.curr.met

Granule 2:
ESDT Short Name : AM1ATTF
Version : 001
Multi File granule : N
File Name :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.next
Met file :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.next.met

Granule 3:
ESDT Short Name : AM1ATTF
Version : 001

Multi File granule : N
File Name :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.next_1
Met file :
$ECS_HOME/<MODE>/CUSTOM/data/DPS/AM1_DEFINITIVE_ATT.fdd.next_1.met

2 PRs need to be created, one for Profile 1 and one for Profile 2. Create the PR for Profile 2 first (it runs on earlier data) and then create the PR for Profile 1.

The time frame for Production request editor for Profile 2 is 1998 June 30 02:00:00 to 1998 June 30 04:00:00

The time frame for Production request editor for Profile 1 is 1998 June 30 04:00:00 to 1998 June 30 06:00:00

Note: The Data files and tar files are delivered only to the Science Data Server and cannot be directly accessed by SSIT. In order to insert these files they must be moved to the SSIT server manually.

26.18.5.3 DPREP Step1 profile1 PCF

PRODUCT INPUT FILES
##
(**Initial Construction Record**:)
1000|P0420004AAAAAAAAAAAAAA98300080000000.PDS|/net/p0drg01/dss_amass/testdata/instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<A) insert UR here>||2
(**Current 2 hour Data File**)
1000|P0420004AAAAAAAAAAAAAA98300080000001.PDS|/net/p0drg01/dss_amass/testdata/instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<B) insert UR here>||1
:
(**Future 2 hour Construction Record**)

1010|P0420004AAAAAAAAAAAAAA98300100000000.PDS|/net/p0drg01/dss_amass/testdata/instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<C)
insert UR here>||2
(**Future 2 hour Data File**)
1010|P0420004AAAAAAAAAAAAAA98300100000001.PDS|/net/p0drg01/dss_amass/testdata/instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<D)
insert UR here>||1

The last ephemeris/attitude data sets generated. **(Found in PGE template)**

Ephemeris (Toolkit native format) **(Previous Time Range)**
1500|EOS_AM1_Ephemeris.21206.step1b.eph|/home/cmts1/DPREP/demooutput||<E) insert UR here>||1
Attitude (Toolkit native format)
1502|EOS_AM1_Attitude.21206.step1.att|/home/cmts1/DPREP/demooutput||<F) insert UR here>||1

```
:
?   PRODUCT OUTPUT FILES
##############################################################################
```
(Current Time Range)
```
# ----------------------------------------------------------------------
# Toolkit attitude datasets (output of EcDpPrAm1EdosAncillary).
# ----------------------------------------------------------------------
1001|EOS_AM1_Attitude.21208.step1.att|/home/cmts1/DPREP/demooutput||||1
1100|EOS_AM1_Attitude.21208.step1.hdf|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Toolkit ephemeris datasets (output of EcDpPrAm1EdosAncillary).
# ----------------------------------------------------------------------
1102|EOS_AM1_Ephemeris.21208.hdf|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Gap-filled HDF ephemeris dataset (output of EcDpPrAm1EphemerisGapFill).
# ----------------------------------------------------------------------
1151|EOS_AM1_Ephemeris.21208.step1b.eph|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Gap-filled Toolkit ephemeris dataset (output of
# EcDpPrAm1FormatNativeEphemeris).
# ----------------------------------------------------------------------
1152|EOS_AM1_Ephemeris.21208.step1b.hdf|/home/cmts1/DPREP/demooutput||||1
:
SUPPORT OUTPUT FILES
##############################################################################
10100|LogStatus|/home/cmts1/DPREP/logs||||1
10101|LogReport|/home/cmts1/DPREP/logs||||1
10102|LogUser|/home/cmts1/DPREP/logs||||1
10103|TmpStatus|||||1
10104|TmpReport|||||1
10105|TmpUser|||||1
10110|MailFile|||||1
:
?   USER DEFINED RUNTIME PARAMETERS
##############################################################################
999|No Previous Data Set; 1=true, 0=false.|0          (Profile 1 set to 0, Profile 2 set to 1)
# ----------------------------------------------------------------------
# Toolkit version for which this PCF was intended.
# DO NOT REMOVE THIS VERSION ENTRY!
# ----------------------------------------------------------------------
10220|Toolkit version string|DAAC B.0 TK5.2.4     (Caution:  May need to change to higher
level Toolkit)
END
```

26.18.5.4 DPREP Step1 profile2 PCF (Profile 2 is start up PCF)

```
PRODUCT INPUT FILES
##############################################################################
1000|P0420004AAAAAAAAAAAAAA98300060000000.PDS|/net/p0drg01/dss_amass/testdata/
instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<A) insert UR here>||2
1000|P0420004AAAAAAAAAAAAAA98300060000001.PDS|/net/p0drg01/dss_amass/testdata/
instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<B) insert UR here>||1
1010|P0420004AAAAAAAAAAAAAA98300080000000.PDS|/net/p0drg01/dss_amass/testdata/
instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<C) insert UR here>||2
1010|P0420004AAAAAAAAAAAAAA98300080000001.PDS|/net/p0drg01/dss_amass/testdata/
instrument_data/AM1/EDOS_HK_ANC/EDOSL0_ANC_PDS_0002||<D) insert UR here>||1
# ----------------------------------------------------------------------
# The last ephemeris/attitude data sets generated.
# ----------------------------------------------------------------------
# Ephemeris (Toolkit native format)
#1500|EOS_AM1_Ephemeris.prev.eph|.||<E) insert UR here>||1
# Attitude (Toolkit native format)
#1502|EOS_AM1_Attitude.21206.step1.att|/home/cmts1/DPREP/output||<F) insert UR here>||1
:
?  PRODUCT OUTPUT FILES
##############################################################################
# ----------------------------------------------------------------------
# Toolkit attitude datasets (output of EcDpPrAm1EdosAncillary).
# ----------------------------------------------------------------------
1001|EOS_AM1_Attitude.21206.step1.att|/home/cmts1/DPREP/demooutput||||1
1100|EOS_AM1_Attitude.21206.step1.hdf|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Toolkit ephemeris datasets (output of EcDpPrAm1EdosAncillary).
# ----------------------------------------------------------------------
1102|EOS_AM1_Ephemeris.21206.hdf|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Gap-filled HDF ephemeris dataset (output of EcDpPrAm1EphemerisGapFill).
# ----------------------------------------------------------------------
1151|EOS_AM1_Ephemeris.21206.step1b.eph|/home/cmts1/DPREP/demooutput||||1
# ----------------------------------------------------------------------
# Gap-filled Toolkit ephemeris dataset (output of EcDpPrAm1FormatNativeEphemeris).
# ----------------------------------------------------------------------
1152|EOS_AM1_Ephemeris.21206.step1b.hdf|/home/cmts1/DPREP/demooutput||||1
?  SUPPORT OUTPUT FILES
##############################################################################
10100|LogStatus|/home/cmts1/DPREP/logs||||1
10101|LogReport|/home/cmts1/DPREP/logs||||1
10102|LogUser|/home/cmts1/DPREP/logs||||1
10103|TmpStatus|||||1
10104|TmpReport|||||1
```

```
10105|TmpUser|||||1
10110|MailFile|||||1
:
?   USER DEFINED RUNTIME PARAMETERS
##########################################################################
999|No Previous Data Set; 1=true, 0=false.|1
# ------------------------------------------------------------------------
# Toolkit version for which this PCF was intended.
# DO NOT REMOVE THIS VERSION ENTRY!
# ------------------------------------------------------------------------
10220|Toolkit version string|DAAC B.0 TK5.2.4
?   END
```

26.18.5.5 DPREP Step2 profile1 PCF

```
PRODUCT INPUT FILES
##############################################################################
1000|AM1_DEFATT_080000_212_1997_01.FDD|/home/cmts1/DPREP/am1defatt||<A) insert
UR here>||1
:
1010|AM1_DEFATT_100000_212_1997_01.FDD|/home/cmts1/DPREP/am1defatt||<B) insert
UR here>||1
# ----------------------------------------------------------------------
# The last ephemeris/attitude data sets generated.
# ----------------------------------------------------------------------
# Attitude (Toolkit native format)
1502|EOS_AM1_Attitude.21206.step2.att|/home/cmts1/DPREP/demooutput||<C) insert UR
here>||1
:
#1400|AM1EPHMH.mcf|.||||1
1401|AM1ATTHF.MCF|/home/cmts1/DPREP/mcf||||1
#1402|AM1EPHMN.mcf|.||||1
1403|AM1ATTNF.MCF|/home/cmts1/DPREP/mcf||||1
10250|MCF|||||1
10252|GetAttr.temp|||||1
10254|MCFWrite.temp|||||1
:
10501|EOS_AM1_Ephemeris.21208.step1b.eph|/home/cmts1/DPREP/demooutput||||2
10501|EOS_AM1_Ephemeris.21210.step1b.eph|/home/cmts1/DPREP/demooutput||||1
10502|INSERT_ATTITUDE_FILES_HERE|||||1
:
?  PRODUCT OUTPUT FILES
##############################################################################
:1001|EOS_AM1_Attitude.21208.step2.att|/home/cmts1/DPREP/demooutput||||1
1100|EOS_AM1_Attitude.21208.step2.hdf|/home/cmts1/DPREP/demooutput||||1
:
?  USER DEFINED RUNTIME PARAMETERS
##############################################################################
:
999|First Mission Data Set; 1=true, 0=false.|0
:10220|Toolkit version string|DAAC B.0 TK5.2.4
:END
```

26.18.5.6 DREP Step2 profile2 PCF

```
PRODUCT INPUT FILES
##############################################################################
```

```
1000|AM1_DEFATT_060000_212_1997_01.FDD|/home/cmts1/DPREP/am1defatt||<A) insert
UR here>||1
:
1010|AM1_DEFATT_080000_212_1997_01.FDD|/home/cmts1/DPREP/am1defatt||<B) insert
UR here>||1
# ----------------------------------------------------------------------
# The last ephemeris/attitude data sets generated.
# ----------------------------------------------------------------------
# Attitude (Toolkit native format)
1502|EOS_AM1_Attitude.21206.step1.att|/home/cmts1/DPREP/demooutput||<C) insert UR
here>||1
:
1400|AM1EPHMH.mcf|.||||1
1401|AM1ATTHF.MCF|/home/cmts1/DPREP/mcf||||1
1402|AM1EPHMN.mcf|.||||1
1403|AM1ATTNF.MCF|/home/cmts1/DPREP/mcf||||1
10250|MCF|||||1
10252|GetAttr.temp|||||1
10254|MCFWrite.temp|||||1
:
10501|EOS_AM1_Ephemeris.21206.step1b.eph|/home/cmts1/DPREP/demooutput||||2
10501|EOS_AM1_Ephemeris.21208.step1b.eph|/home/cmts1/DPREP/demooutput||||1
10502|INSERT_ATTITUDE_FILES_HERE|||||1
:
?   PRODUCT OUTPUT FILES
##########################################################################
1001|EOS_AM1_Attitude.21206.step2.att|/home/cmts1/DPREP/demooutput||||1
1100|EOS_AM1_Attitude.21206.step2.hdf|/home/cmts1/DPREP/demooutput||||1
:
?   USER DEFINED RUNTIME PARAMETERS
##########################################################################
:999|First Mission Data Set; 1=true, 0=false.|1
:
10220|Toolkit version string|DAAC B.0 TK5.2.4
:
?   END
```

26.18.5.7 Setups for DPREP

```
set path = ( /usr/ecs/TS1/CUSTOM/TOOLKIT/toolkit/bin/sgi_daac_f90 $path )
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step1DP21206.PCF
setenv PGSMSG /home/cmts1/DPREP/msg
setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step1DP21208.PCF
setenv PGSMSG /home/cmts1/DPREP/msg
```

setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step1DP21210.PCF
setenv PGSMSG /home/cmts1/DPREP/msg
setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step2DP21206.PCF
setenv PGSMSG /home/cmts1/DPREP/msg
setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
setenv PGS_PC_INFO_FILE /home/cmts1/DPREP/pcf/Step2DP21208.PCF
* setenv PGSMSG /home/cmts1/DPREP/msg
* setenv PGSHOME /usr/ecs/TS1/CUSTOM/toolkit
Note: The **setenv** parameters are used for the **Command Line PGE** test. Only the **(*)** are used for **PDPS PGE** runs.

26.18.6 FDD Ephemeris Reprocessing Due To QA Failure

The following events occur when FDD-supplied ephemeris fails QA checking:

11 A subscription is placed on ESDT AM1EPHNF for replacement data (ESDT AM1EPHF) when QaPercentMissingData or QaOutOfBoundsData is greater than or equal to 1.

12 The AM1 DPREP FDD ephemeris processor (“RepEph” or “Step 3” DPREP PGE) detects a data gap or an out-of-bounds data point in the FDD-supplied ephemeris timeline (supplied through ESDT ShortName AM1EPHF). Assuming that no fatal errors are otherwise encountered, the PGE exits with status code 216, a success condition that indicates that the FDD-supplied ephemeris data set must be replaced by a re-delivery of the *same* data set. The *QaPercentMissingData* or QaOutOfBoundsData metadata has been set to a value greater than or equal to 1.

13 A subscription triggers when the DPREP output data set (i.e. AM1EPHNF) that is generated in step 2 is inserted into the archive. The subscription triggers because of *QaPercentMissingData* or QaOutOfBoundsData being greater than or equal to 1.

14 The GDAAC operator receives an E-mail notification indicating that the subscription has triggered. The universal reference (UR) of the AM1EPHNF granule that triggered the subscription is provided in the E-mail.

15 The GDAAC operator uses the database ID imbedded within the UR to identify the AM1EPHNF granule in the archive using *EcCoDbViewer*. Retrieve the start and end times of this granule.

16 The GDAAC operator telephones the FOT (telephone number 301-614-5431) and requests an FDD ephemeris data set (ESDT ShortName AM1EPHF) be sent for the time period spanned by the start and end times determined in step 5. The operator explicitly states what the replacement granule’s start and end time must be, to the second. The replacement ephemeris data set will be sent to both the GDAAC and the LDAAC.

17 The GDAAC operator notifies LDAAC operations (telephone number 757-864-9197) that a new AM1EPHF ephemeris data set is going to arrive for the time period determined in step 5.

18 The GDAAC and LDAAC operators schedule the AM-1 DPREP FDD ephemeris processor (“RepEph” or “Step 3” DPREP PGE) to process the time interval determined

in step 5. If data replacement occurs on the first granule following a period of data dropout, Profile 2 processing must be scheduled. Otherwise, Profile 1 can be scheduled to run.

26.18.6.1 Boot-up of EDOS Ephemeris Processing

AM-1 DPREP EDOS ephemeris processor ("AM1Eph" or "Step 1" DPREP) Profile 2 requires a special procedure in order to achieve boot-up processing at the start of the mission and following periods of data dropout. The required steps follow:

1. Wait for the first EDOS-supplied AM1ANC data set that follows the interval of data dropout to be ingested.
2. Use the *EcCoDbViewer* archive browser to determine the start and end time of the granule.
3. Call the FOT (telephone number 301-614-5431) and ask to speak with the on-line engineer.
4. Ask the on-line engineer to provide the orbit number at the granule start time determined in step 2.
5. Telephone LDAAC operations (telephone number 757-864-9197). Pass-on the orbit number determined in step 4 to the operator so the LDAAC can proceed with steps 6 through 9. This avoids having both the LDAAC and the GDAAC perform steps 1 through 4.
6. In directory /usr/ecs/OPS/CUSTOM/data/DPS, locate ODL file PGE_AM1Eph*VVVVV*02.odl. *VVVVV* is the version number of the operational AM-1 DPREP.
7. Edit this file using vi. Locate logical ID 998 (*PGE_PARAMETER_NAME InitialOrbitNumber*) within the ODL file and insert the orbit number provided by the FOT on-line engineer into the line beginning with *PGE_PARAMETER_DEFAULT*. This step requires *allmode* privileges in order to edit the ODL file.
8. Register the "AM1Eph" AM-1 DPREP.
9. Schedule "AM1Eph" AM-1 DPREP, Profile 2 to process the interval given by the start and end times determined in step 2.

26.18.6.2 FDD Replacement Ephemeris Processing

The following events occur in FDD replacement ephemeris processing:

1. A subscription is placed on ESDT AM1EPHN0 for replacement data (ESDT AM1EPHF) when QaPercentMissingData is greater than or equal to 1.
2. The AM1 DPREP EDOS ephemeris processor ("AM1Eph" or "Step 1" DPREP PGE) detects a long data gap in the EDOS-supplied ephemeris timeline (supplied through ESDT ShortName AM1ANC). Assuming that no fatal errors are otherwise encountered, the PGE exits with status code 216, a success condition that indicates the replace ephemeris condition has been detected. The *QaPercentMissingData* metadata is set to a value greater than or equal to 1, depending on the size of the data gap that is detected.

3. A subscription triggers when the DPREP output data set (i.e. AM1EPHN0) that is generated in step 2 is inserted into the archive. The subscription triggers because of *QaPercentMissingData* being greater than or equal to 1.
4. The GDAAC operator receives an E-mail notification indicating that the FDD replacement ephemeris data set subscription has triggered. The universal reference (UR) of the AM1EPHN0 granule that triggered the subscription is provided in the E-mail.
5. The GDAAC operator uses the database ID imbedded within the UR to identify the AM1EPHN0 granule in the archive using *EcCoDbViewer*. Retrieve the start and end times of this granule.
6. Given the start and end granules times, the GDAAC operator derives the replacement time range. The procedure will be demonstrated by example. If the 2-hour AM1EPHN0 granule ideally spans 22h – 24h of day 2000-06-07, the replacement ephemeris granule time span is

   Start time = 2000-06-07 22:00:00.000
   End time = 2000-06-07 23:59:59.000

   Replacement data starts on the hour and ends one second prior to the start of the subsequent 2-hour granule.
7. The GDAAC operator telephones the FOT (telephone number 301-614-5431) and requests an FDD replacement data set (ESDT ShortName AM1EPHF) be sent for the time period spanned by the start and end times determined in step 6. The operator explicitly states what the replacement granule's start and end time must be, to the second. The replacement ephemeris data set will be sent to both the GDAAC and the LDAAC.
8. The GDAAC operator notifies LDAAC operations (telephone number 757-864-9197) that a replacement ephemeris data set (AM1EPHF) is going to arrive for the time period determined in step 6.
9. The GDAAC and LDAAC operators schedule the AM-1 DPREP replacement ephemeris processor ("RepEph" or "Step 3" DPREP PGE) to process the time interval determined in step 6. If data replacement occurs on the first granule following a period of data dropout, Profile 2 processing must be scheduled. Otherwise, Profile 1 can be scheduled to run.

26.18.6.3 FDD Attitude Reprocessing Due To QA Failure

The following events occur when FDD-supplied attitude fails QA checking:

a. A subscription is placed on ESDT AM1ATTNF for replacement data (ESDT AM1ATTF) when QaPercentMissingData or QaOutOfBoundsData is greater than or equal to 1.
b. The AM1 DPREP FDD attitude processor ("FddAtt" or "Step 2" DPREP PGE) detects a data gap or an out-of-bounds data point in the FDD-supplied attitude timeline (supplied through ESDT ShortName AM1ATTF). Assuming that no fatal errors are otherwise encountered, the PGE exits with status code 216, a success condition that indicates that the FDD-supplied attitude data set must be replaced by a re-delivery of the *same* data set. The *QaPercentMissingData* or QaOutOfBoundsData metadata has been set to a value greater than or equal to 1.
c. A subscription triggers when the DPREP output data set (i.e. AM1ATTNF) that is generated in step 2 is inserted into the archive. The subscription triggers because of *QaPercentMissingData* or QaOutOfBoundsData being greater than or equal to 1.

d. The GDAAC operator receives an E-mail notification indicating that the subscription has triggered. The universal reference (UR) of the AM1ATTNF granule that triggered the subscription is provided in the E-mail.
e. The GDAAC operator uses the database ID imbedded within the UR to identify the AM1ATTNF granule in the archive using *EcCoDbViewer*. Retrieve the start and end times of this granule.
f. The GDAAC operator telephones the FOT (telephone number 301-614-5431) and requests an FDD attitude data set (ESDT ShortName AM1ATTF) be sent for the time period spanned by the start and end times determined in step 5. The operator explicitly states what the replacement granule's start and end time must be, to the *millisecond*. The replacement attitude data set will be sent to both the GDAAC and the LDAAC.
g. The GDAAC operator notifies LDAAC operations (telephone number 757-864-9197) that a new AM1ATTF attitude data set is going to arrive for the time period determined in step 5.
h. The GDAAC and LDAAC operators schedule the AM-1 DPREP FDD attitude processor ("FddAtt" or "Step 2" DPREP PGE) to process the time interval determined in step 5. If data replacement occurs on the first granule following a period of data dropout, Profile 2 processing must be scheduled. Otherwise, Profile 1 can be scheduled to run.

26.19 PGE Chaining

26.19.1 Chaining PGE's

1. Create PRs (so that DPRs) for the PGEs to be chained.

   This can be done by using PR Editor. Follow the same procedure as creating independent PR.

   A few points need to be noticed. Let's say among the chained PGEs, the output of PGE A will be the input of PGE B.1) In ESDT odl for this shared granule, "DYNAMIC_FLAG" has to be set to "I", i.e., dynamic internal. 2) First create PR for PGE A, then for PGE B. Otherwise PGE B PR may not be able to be generated.

2. Create the plan for a bunch of PRs which are chained.

   In Work Bench GUI, 1) pull down "file" menu and select "new" to create the new plan; 2) highlight all PRs that are chain by clicking on their names on "unscheduled" area of Production Request area; 3) click schedule button to schedule these PRs.

3. Activate the plan.

   In the Workbench GUI, click "activate" button, a GUI will pop up to ask for saving the plan. Answer "yes". Then another GUI will pop up to confirm whether to really activate the plan. Answer "yes" and the lowest level of DPR(s) in the chain will kick off.

   In the pdps database, the PlDataProcessingRequest table is where the PRs are successfully generated, the "completionState" for all DPRs in the chain are "NULL". When the plan is successfully activated, the "CompletionState" for lowest level of DPR(s) is changed from "NULL" to "STARTED". The high level of DPR(s) in the chain is changed from "NULL" to "CQ_HOLD". Eventually, the low level of DPR(s) finish so

that the input for high level of DPR(s) become available, Then the high level DPR(s) kick off and the "CompletionState" then changes from "CQ_HOLD" to "STARTED".

26.20 Updating the Orbit Model

26.20.1 Introduction to Updating the Orbit Model

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. 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.

26.20.2 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.
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.

7 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!

26.21. Learn more about SSI & T

26.21.1 References:

PDPS Home Page: http://dmserver.gsfc.nasa.gov/ecsdev/relb/pdps/index.html

` URL for ECS Project Training Material Volume 16: SSI&T December 1997:

http://m0mss01.ecs.nasa.gov/smc/dc master.html

MISR Science Data Processing Software Test Plan, Volume 2, Detailed Procedures and Facilities Version 2.0, Part 1 (PGE 1) June 1998.

http://dmserver.gsfc.nasa.gov/relbit/relbit.html

SV DOC

REPOSITORY

Home

Drop Build Plans

Acceptance Test Plan

System Verifcation Test Plan

Access Database

Release B Testdata

Site Install and Checkout Test

End To End Test Procedures

Goddard DAAC M&O Status

VATC Status Page

ECS TEST
PAGES

Advertising Service (VATC)
User Registration Tool
(VATC)
V0 Web Client (VATC TS2)
V0 Web Client (DAAC
MODE TS2)
V0 Web Client (DAAC
MODE TS1)
V0 Web Client (DAAC
MODE OPS)

ECS TOOLS

EP7
RTM
CCR
EDHS
DDTS
Network Status Page
ECS Newsroom Server
Configuration Management
Release B Integration
ECS Telephone & Email Dir

ISO 9001

Business Manual Tab

Job Description Tab

Organization Charts Tab

Process Directives Tab

Training Tab

Miscellaneous Tab

Frequently Asked questions

EDHS

Raytheon Company

ESDT Basics http://dmserver.gsfc.nasa.gov/esdt/EsdtSection1/index.html

GDAAC Directory for SSI&T:http://gsfcsrvr8.gsfcmo.ecs.nasa.gov/SSIT/

The ESDT Process (updated for drop 4) by Karl W. Cox, 22 December 1997

DCE Cell Manager Common User Tasks provided by IDG, February 6, 1998

1 Tools and Techniques for Diagnosing Potential DCE Problems

MODIS - Science Data Processing Software Release 4 System Description

SDST-104, May 19, 1998

ECSINFO: http://ecsinfo.hitc.com/iteams/Science/science.html

PDPS howto are located on the EDF machines at: **/home/PDPS/docs/**

PDPS Web Page: **http//dmserver.gsfc,nasa.gov/ecsdev/relb/pdps/index.html**

For Troubleshooting or use the following EDF machines:

PDPS Troubleshooting Techniques are located on the EDF machines at **: /home/PDPS/troubleshooting/**

DPREP README files located at **:/usr/ecs/TS1/CUSTOM/data/DPS/**

DPREP binary located: **:/usr/ecs/TS1/CUSTOM/bin/DPS/**

26.21.2 Server Node Names Convention:

The naming convention is as follows:

Machine names are defined to be equivalent to the network hostname of the machine. Network hostnames are limited to eight characters. On ECS we are now formatting these hostnames as svcimnni

Wheres : Site

- g – GSFC
- e – EDC
- l – LaRC
- n – NSIDC
- a – ASF
- j – JPL
- p- PVC
- t – VATC

v : Version

- 0 -- Release 4 At-Launch COTS design
- 1 -- Stood for B.1 COTS design; OBE, but still used in VATC
- s -- Used for special SSI&T machines set up at GSFC and EDC

ci : Hardware Configuration Item

sp -- Science Processing (SPRHW)

ai -- Algorithm Integration and Test (AITHW)

aq -- Algorithm Quality Assurance (AQAHW)

pl -- Planning (PLNHW)

ms -- Management Subsystem (MSSHW)

cs -- Communications Subsystem (CSSHW)

in -- Interface (INTHW)

dm -- Data Management (DMGHW)

dr -- Data Repository (DRPHW)

ac -- Access Control Management (ACMHW)

ic -- Ingest Client (ICLHW)

wk -- Working Storage (WKSHW)

di -- Distribution (DIPHW)

as -- ASTER (ASTHW) [Occurs only at EDC]

te -- Test Equipment

m : Manufacturer

s -- Sun

g -- SGI

h -- HP

x -- X Terminal

p -- PC

nn : One-up number (01, 02, et cetera -- should be unique for the CI)

i : Interface type

<null> -- Production network

u -- User network

h -- HiPPI

Note that the machine name leaves off the last letter (the interface); hence, we generally refer to machines as "g0spg01", vice "g0spg01h". A machine may have multiple interfaces -- production, user, and HiPPI. So a single machine may show

up in network documentation multiple times (g0spg01, g0spg01h, g0spg01u).

26.21.3 A Handy Alias file to use while conducting SSI&T:

p0spg01{emcleod}51: **alias**
+ pushd
- popd
More more !* |grep -v "Msg: Caught dce error: No more bindings (dce / rpc)" |grep -v "MsgLink :0 meaningfulname :EcAgManager::Recovery" |grep -v "MsgLink :0 meaningfulname :DsShSRequestRealSetStateSettingState" |grep -v "Command 1/1 execution complete"
cdstagebin cd /ecs/formal/STAGE/DSS/bin/sun5.5
dbg debugger -bg NavajoWhite -fn 12x24 !* &
dbs /home/jzhuang/bin/dbrowser-syb &
disp setenv DISPLAY !*
mgr DpAtMgr ConfigFile /usr/ecs/TS1/CUSTOM/cfg/DpAtMG.CFG ecs_mode TS1&
ops cd /usr/ecs/OPS/CUSTOM
ts1 cd /usr/ecs/TS1/CUSTOM
xslq_autosys isql -Uautosys -Pautosys -Sp0sps06_srvr
xsql_css isql -Ucss_role -Pwelcome -Sp0ins01_srvr
xsql_dss isql -UsdsrvApp -Pwelcome -Sp0acg01_sqs222_srvr
xsql_ios isql -Uios_role -Pwelcome -Sp0ins02_srvr
xsql_pdps isql -UpdpsUsers -Pwelcome -Sodyssey_srvr

 # alias for browser
Note: On Performance Verification Center (PVC), dbbrowser has to originate from workstation ODYSSEY to execute alias db_pdps to reach PDPS DB on p0pls02.
alias db_dss '/home/opscm/dbr/dbrowser-syb -UsdsrvApp -Pwelcome -Sp0acg01_sqs222_srvr &'
alias db_ios '/home/opscm/dbr/dbrowser-syb -Uios_role -Pwelcome -Sp0ins02_srvr &'
alias db_css '/home/opscm/dbr/dbrowser-syb -Ucss_role -Pwelcome -Sp0ins01_srvr &'
alias db_autosys '/home/opscm/dbr/dbrowser-syb -Uautosys -Pautosys -Sp0sps06_srvr &'
alias db_pdps '/home/opscm/dbr/dbrowser-syb -UpdpsUsers -Pwelcome -Sp0pls02_srvr &'
alias db_ing '/home/opscm/dbr/dbrowser-syb -UEcInPolling -P3nWK0fG1 -Sp0icg01_srvr &'
alias db_stmgt '/home/opscm/dbr/dbrowser-syb -UEcDsStFtpDisServer -PS71Oq4y3 -Sp0icg01_srvr &'

ls -laF look at root and .cshrc, .alias

26.21. 4 HOWTO_SSIT HELPFUL NOTES

Xterm format to bring up xterm windows for servers all at once

This list of xterm identifiers should be assigned a file name and placed into your home directory. Then invoke the file name when you want to create the entire list of xterms.

Example of filename: - xterm_pls

```
xterm -sb -sl 10000 -fg green -bg black -name "Resource Editor" &
xterm -sb -sl 10000 -fg green -bg black -name "Resource Model" &
xterm -sb -sl 10000 -fg green -bg black -name "Production Request Editor" &
xterm -sb -sl 10000 -fg green -bg black -name "Planning Workbench" &
xterm -sb -sl 10000 -fg green -bg black -name "Database" &
xterm -sb -sl 10000 -fg green -bg black -name "Planning Timeline" &
xterm -sb -sl 10000 -fg green -bg black -name "Logs" &
xterm -sb -sl 10000 -fg green -bg black -name "ECS Assist" &
xterm -sb -sl 10000 -fg green -bg black -name "g0sps06" &
xterm -sb -sl 10000 -fg green -bg black -name "g0ais01" &
xterm -sb -sl 10000 -fg green -bg black -name "g0spg01" &
xterm -sb -sl 10000 -fg green -bg black -name "g0drg01" &
xterm -sb -sl 10000 -fg green -bg black -name "g0pls02" &
# setenv ECS_HOME /usr/ecs
# setenv MODE TS2
```

26.21.5 howto_setup_orbits_and_pathmaps

ORBITS & PATHMAPS
The Path is a an orbit swath, defined for Landsat-7 ("WRS"),
which MISR uses in its processing.
Because the earth rotates under it, the path the satellite
traverses for its next orbit is not path 2, but path 17.
This mapping of orbit number to path number is found in the
PATHMAP ODL file.
In that file ABSOLUTE_PATH is what we call orbit number here,
and MAPPED_PATH is the path number.
Now, this mapping is fixed, and never changes.
What does change is the time each orbit starts.
This is because the orbit may drift and be subject to
maneuvers.
Periodically (say, every 2 weeks), the Flight Dynamics
Facility (FDF) at GSFC issues a new Orbit Start Time,
with corresponding Orbit Number.
When this happens, the PDPS ORBIT ODL must be updated,
with a new ORBIT_MODEL object, containing the new
ORBIT_START and corresponding ORBIT_NUMBER.
The new ORBIT_PATH_NUMBER is determined manually by

the operator, using the lookup table in the PATHMAP ODL file.
1. Example (MISR PGE7 test data)
FDF issues a bulletin stating that imaginary platform MPGE7
orbit number 27 starts at 14:37:39Z 07-Jan-96.
SSIT operator receives the bulletin, looks up in
the corresponding PATHMAP_WRS7.odl file to find that
ABSOLUTE_PATH XX corresponds to MAPPED_PATH XX.
NEED TO FIX THIS
The SSIT operator then creates a new ORBIT_MODEL object in the
ORBIT ODL file as follows:

```
OBJECT = ORBIT_MODEL
    CLASS = 2
    ORBIT_NUMBER = 27
    ORBIT_PERIOD = "SECS=5932"
    ORBIT_START = "01/07/1996 14:37:39Z"
    ORBIT_PATH_NUMBER = 90
END_OBJECT = ORBIT_MODEL
```

****

26.21.6 howto_register_pge

```
# First prepare PGE and ESDT ODL files
# Update database
xterm -sb -sl 256 -bg maroon -fg "papaya whip" -cr "papaya whip" -fn
"*l*s*type*b*r*140
*" -T 'SSIT: Science Metadata Database Update' -n 'Science Metadata Update' -e
/usr/ecs
//OPS/CUSTOM/bin/DPS/EcDpAtDefinePGE &
# Update performance info
/usr/ecs//OPS/CUSTOM/bin/DPS/EcDpAtOpDbGui ConfigFile
/usr/ecs//OPS/CUSTOM/cfg/EcDpAtOp
DbGui.CFG ecs_mode OPS &
```

****

26.21.7 howto_register_dpr

```
##############################################################################
# Make sure STMGT and SDSRV are up
##############################################################################
# Use either ECS Assist on p0acs03 *and* p0drg01,
# Alternatively, use ps:
p0acs03:> ps -ef | grep EcDsScienceDataServer | grep TS1
  sdsrv 24410     1  0 10:17:57 pts/3    0:33
/usr/ecs//TS1/CUSTOM/bin/DSS/EcDsScience
DataServer ConfigFile /usr/ecs//TS1/CUS
p0drg01:> ps -ef | grep  Server | grep TS1 | grep ConfigFile
  stmgt  3786     1  0   Dec 30 ?      10:39
```

/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsStStaging
DiskServer ConfigFile /usr/ecs/TS1/CUS
  stmgt 3803   1 0   Dec 30 ?   10:31
/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsStStaging
MonitorServer ConfigFile /usr/ecs/TS1/
  stmgt 3770   1 0   Dec 30 ?   16:18
/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsStArchive
Server ConfigFile /usr/ecs/TS1/CUSTOM/
  stmgt 3815   1 0   Dec 30 ?   10:59
/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsStFtpDisS
erver ConfigFile /usr/ecs/TS1/CUSTOM/c
If any of these are not present, that server is down.
You must arrange to have it up before continuing
****

26.21.8 Using Production Request Editor and Planning Workbench

PRE:
telnet p0pls01, cmops, dce_login
ops, cd utilities,
then invoke EcplStartPRE_IF OPS 3 &
PWB: telnet p0pls01, cmops, dce_login
ops, cd utilities, invoke EcPlSlayAll prior to using PWB.
      then invoke EcPlStartAll OPS 3 & or 2
      times used for plan activation: 01/01/1990 time 00:00:00
AutoSys- To Monitor DPS , telnet to: p0sps06_srvr
##########

26.21.9 howto_Make a Production Request

##########
rlogin p0pls01 -l pls
dce_login
awhitele
cd /usr/ecs/TS1/CUSTOM/utilities
set path = (/usr/bin)
source .buildrc
setenv DISPLAY mojave:0.0
Make sure no one else is running PREditor, then run it
/usr/ucb/ps -auxwww | grep -i pre | grep TS1
EcPlStartPRE_IF TS1 1
Click PREdit
Click PGE, select it, OK

Set time to input, e.g. (CPGE1)
06 12 1997 00 00 00
06 14 1997 00 00 00
****

26.21.10 Additional SSI&T Howto_procedures and Production rules taken from

PDPS tests documented in the following files last updated 01/14/00:

SCF: cd /home/PDPS/docs - look through list of 'howto_s'
production_rules_wp.doc*
Drop3Scen2.txt
HowToCompileAndRunSyntheticPGE
HowToDealWithDPSDeadLocks
HowToFakeSubscriptionNotification
HowToInstallOnComanche
HowToStartAutosysAndViewJobStates
HowToActivateAPlan
HowToDeleteAndRecreateJobs
HowToSetUpPlanning
Tiling.txt
HowToRunProductionStrategyGUI
SpatialQuery.txt
HowToInsertMultiFileGranules
HowToStartDPS
ModisDataList
HowToUseSubscriptionServerDriverToSendSubscriptionNotification
HowToReactivateReplan
VerifiedList
HowToStartQaMonitor
HowToTestFaultRecovery
HowToTestGroundEventJobs
HowToCleanAutoSys
HowToInstall-General
HowToInsertData
MergeList
HowToEnterGroundEvents
HowToTestDynamicMetaDataQuery
HowToReacquireAGranule
ReacquireGranule.sql
HowToAdHocReprocessPRs
HowToTestDatabaseCleanupScript
HowToDoProfiling
HowToRunMulti-GranuleESDTsTest
HowToRunSSITAcquireTool#
HowToRunResourcePlanning
HowToCreateDprepTarFile*HowToRunBROWSETest~

HowToRunBROWSETest
HowToRunSpatial
mailfile
HowToRunDataDay-InterimFilesTest~
HowToTest
HowToInstallPlanning~
HowToRunDataDay-InterimFilesTest
HowToInstallPlanning
HowToRunSSIT
HowToRunOptionalDPRs~
EndToEnd
HowToRunASTER
HowToRunOptionalDPRs
HowToRunTiling~
HowToRunMostRecentGranule~
HowToRunMostRecentGranule
HowToRunTiling
HowToTestAlternateInput~
HowToTestAlternateInput
HowToRunASTER#
HowToRunModis
HowToRunMISRLIKE~
HowToRunModisPlus
HowToRunMISRLIKE
HowToRunASTERRoutine
HowToRunDPREP
HowToRunMostRecentGranule#
HowToRunTiling.odl
HowToTestAlternateInput~
HowToRunASTERRoutine
WhatTestsWhat
HowToRunSSIT
datadepend_dprep
HowToCreateDprepTarFile.old
HowToRunModis.old
HowToRunModisPlus#
HowToRunMRG+MetadataCombo~
HowToRunMRG+MetadataCombo
EndToEnd
HowToTestOutputMFG~
HowToTestOutputMFG
HowToRunSpatial
HowToRunDPREP_Func_Lab~
HowToTestGroundEventJobs
HowToTestDatabaseCleanupScript

HowToRunASTERRoutine
HowToTestDatabaseCleanupScript
HowToRunBROWSETest
HowToRecoverNotifications
LaRC_Synthetic_EndToEnd_Performance Verification Center (PVC)~
LaRC_Synthetic_EndToEnd_Performance Verification Center (PVC)
HowToRunMISRISH
HowToRunMISRSIM
HowToRunMISRLIKE~
HowToRunDataDay-InterimFilesTest
foo
HowToTestAlternateInput
HowToTestAlternateWithInternalDynamics_bad
HowToRunMISRLIKE
xemacs
HowToRunResourcePlanning
HowToTestAlternateWithInternalDynamics*
error.log
HowToRunMRG+MetadataCombo#
hgj_spatial
HowToRunASTER
HowToRunCG+MetadataComboOLD
dead.letter
HowToRunSpatialPad#*
HowToRunMISRLIKE#.NeedAnsToQuesAboutMISRLIKEprofile2
HowToRunSpatialPad~*
HowToRunAIRSLIKE~
HowToRunModis#
HowToRunShortModis
HowToRunShortModis#
HowToRunMulti-GranuleESDTsTest.short~
HowToRunMulti-GranuleESDTsTest.short
HowToCreateDprepTarFile.bh.update.oct8
#HowToRunMulti-GranuleESDTsTest.short#
HowToRunAIRSLIKE
HowToRunASTER1
HowToRunMulti-GranuleESDTsTest.jutout~
HowToRunMulti-GranuleESDTsTest.jutout
HowToRunMISRLIKE
HowToRunASTER#
HowToRunSSITforOnDemand~
HowToRunOnDemandPge~
HowToRunSpatialPad*
HowToRunClosestGranule
HowToRunCG+MetadataCombo

HowToRunModis
HowToRunDPREP_Func_Lab
HowToRunASTERnew.html~
HowToRunMulti-GranuleESDTsTest
HowToRunSSIT.html~
HowToRunResourcePlanning.html~
HowToRunASTER.html~
HowToRunDPREP#
HowToRunModis.html~
HowToRunResourcePlanning.html
HowToRunQAMonitor.html~
HowToRunMISRLIKE.html~
HowToRunModis.html
HowToRunSSIT.html
pdpsTestIndex.html~
TestIndex.html~
HowToRunMulti-GranuleESDTsTest.html~
HowToRunQAMonitor.html
HowToRunMulti-GranuleESDTsTest.html
HowToRunMISRLIKE.html
HowToRunOnDemandPge
HowToDeleteAndRecreateJobs.html
LabIndex.html~
HowToRunASTER.html
SCCS/
LabIndex.html

26.21.11 Technical Notice concerning Leap Second/DPREP

Notice of Change to Toolkit Ephemeris and
Attitude Interpolation

March 15, 1999

Effective with a late patch to Drop 4PY, the time interval over which the Toolkit will interpolate spacecraft ephemeris or attitude is reduced from 121 seconds to 60 seconds. (The normal interval between packets for AM1 is 1.024 seconds, except for FDD replacement ephemeris, which will use 1 second time intervals.)

Impact of this Change

For reasons explained below, the only impact on real data

processing would be to avoid the possibility of a less-than-ideal interpolation.

The impact on I&T would be that test data should always be generated or obtained with packet interval 60 seconds or less. There are a few inputs for the Toolkit test drivers, which are not a deliverable, but are occasionally supplied to Toolkit users, which (to save file space) are set up to work with 120 second packets. These lie in the file "orbsim.in" (which is, as explained, a PDPS add-on which is unsupported software when it is provided to users). To conduct tests without new and annoying failures, if you use this input file for "orbsim", you should edit it to replace "120" by "60" or less, globally. Results will not change significantly from the expected test results.

Rationale for this Change

The SDP Toolkit uses an analytic, cubic polynomial method to interpolate ephemeris that is extremely fast, but which could generate undesirable and spurious variations if applied across too large a time interval, or to data that are not smooth.

DPREP, which will process all incoming ephemeris data, has a robust method for patching gaps up to 60 seconds, based on a least squares quartic fit, component by component, to position and velocity. Although this method is undesirably slow for repeated use within the Toolkit, it is ideal for DPREP, which runs only once on a data set. The existence of any gaps not repairable by DPREP (i.e. > 60 seconds) will trigger procedures to obtain replacement data files from FDD.

It is therefore unwise to allow the Toolkit to interpolate gaps which DPREP cannot fill, so it is being changed. The same limit is set for attitude. We expect no gaps whatever in FDD attitude, and the attitude is not of much use without the ephemeris, so this change was made consistently.

26.22 Using IQ Software to Create Reports

26.22.1 Creating Reports Using IQ Software

ECS no longer plans to offer a Report Generator GUI. Consequently, DAAC operations personnel must use other means to generate various types of reports.
IQ (Intelligent Query) software is a set of commercial off-the-shelf (COTS) products that provides flexible access to the PDPS database from which data for reports can be retrieved. The cost of that flexibility is a somewhat complicated process for initially setting up reports. However, once a particular type of report has been set up, reports can be generated fairly quickly.
The procedure for creating reports using IQ software starts with the assumption that the Production Planner has logged in to the system.

Creating Reports Using IQ Software

NOTE: If using an X-Terminal, it may be necessary to add the following line to the **.Xdefaults** file in the home directory before performing the task for the first time:

```
iqx*background:                    grey
```

NOTE: Commands in Steps 1 through 8 are typed at a UNIX system prompt.

1 Type **rlogin** ***hostname*** refers to the host (e.g., **e0mss21**, **g0mss21**, **l0mss21**, or **n0mss21**) on which GUIs are to be launched during the current operating session. Multiple hostnames can be specified on the same line.

2 Type **setenv DISPLAY** ***clientname*****:0.0** then press the **Return/Enter** key.

- Use either the X terminal/workstation IP address or the machine-name for the ***clientname***.
- When using secure shell, the DISPLAY variable is set just once, before logging in to remote hosts. If it were to be reset after logging in to a remote host, the security features would be compromised.

3 Open another UNIX (terminal) window.

4 Start the log-in to the Applications Server host by typing **/tools/bin/ssh *hostname*** (e.g., **e0mss21**, **g0mss21**, **l0mss21**, or **n0mss21**) in the new window then press the **Return/Enter** key.

- If you receive the message, **Host key not found from the list of known hosts. Are you sure you want to continue connecting (yes/no)?** type **yes** ("**y**" alone will not work).
- If you have previously set up a secure shell passphrase and executed **sshremote**, a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears; continue with Step 5.
- If you have not previously set up a secure shell passphrase; go to Step 6.

5 If a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears, type your ***Passphrase*** then press the **Return/Enter** key.

- Go to Step 7.

6 At the **<*user@remotehost*>'s password:** prompt type your ***Password*** then press the **Return/Enter** key.

7 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

8 Type **cd /usr/ecs/*MODE*/COTS/iqx5** then press **Return/Enter**.

- Change directory to the directory containing the IQ software (directory path may vary from site to site).
- The ***MODE*** will most likely be one of the following operating modes:
- OPS (for normal operation).
- TS1 (for Science Software Integration and Test (SSI&T)).
- TS2 (new version checkout).
- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

9 Type **iqx &** then press **Return/Enter**.

- If the GUIs are not displayed when the command **iqx** is given, try using **./iqx** instead.
- The **iqx IQView** GUI (Figure 26.19.6-1) and either the **iqx Sybase Database Logon** GUI (Figure 26.19.6-2) or **iqx Open IQView** GUI (Figure 26.19.6- 3) are displayed.

— If IQViews have been defined previously, they are listed on the **iqx Open IQView** GUI (Figure 26.19.6-3); otherwise, a list of database tables is displayed.

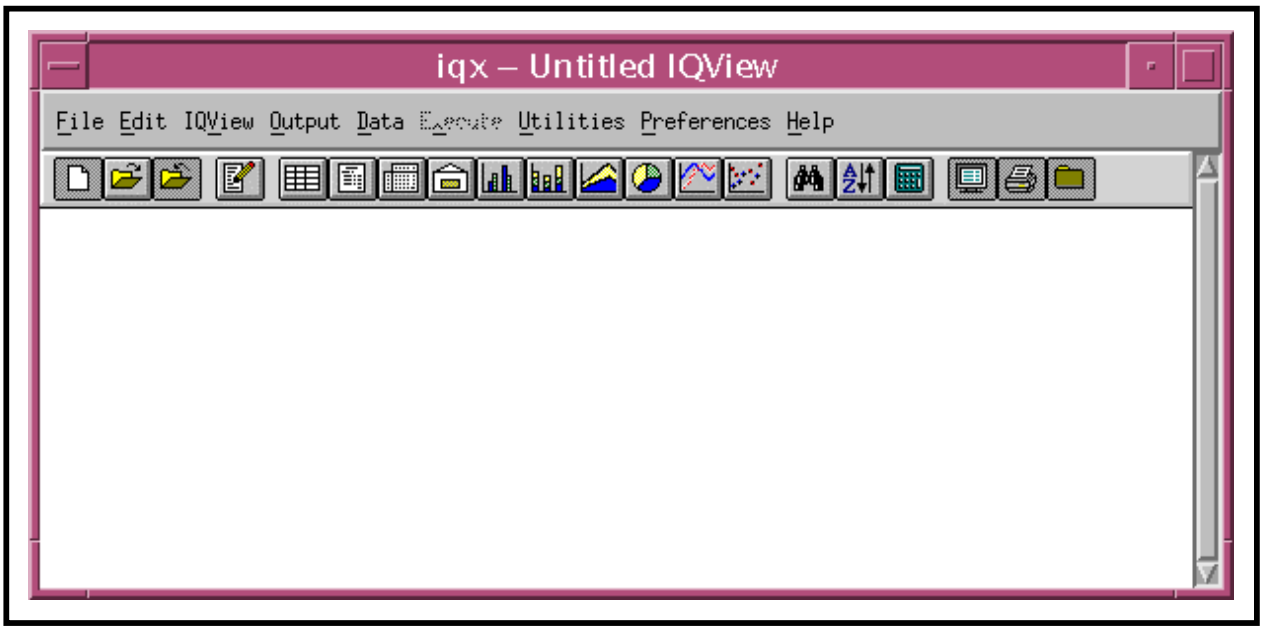


Figure 26.22.1-1. iqx IQView GUI

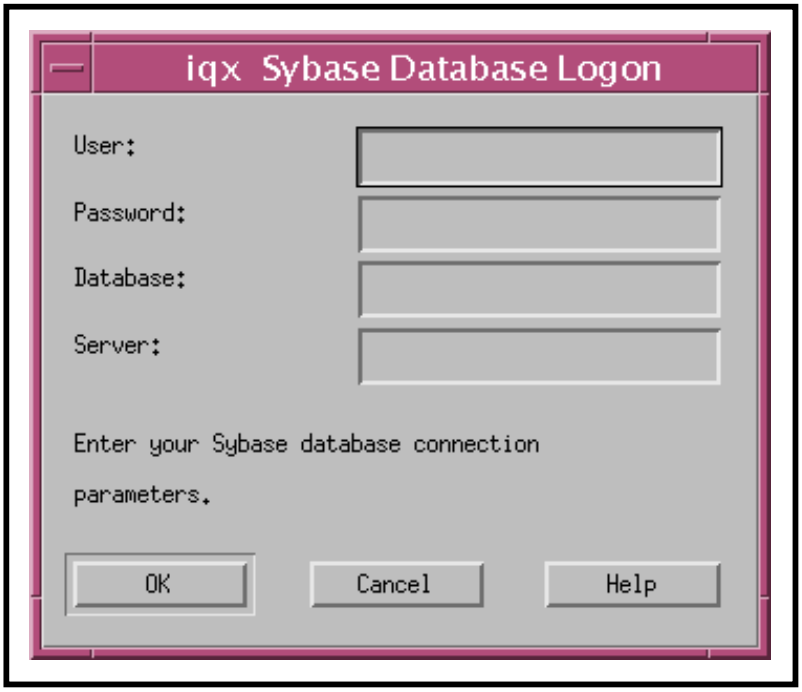


Figure 26.22.1-2. iqx Sybase Database Logon GUI

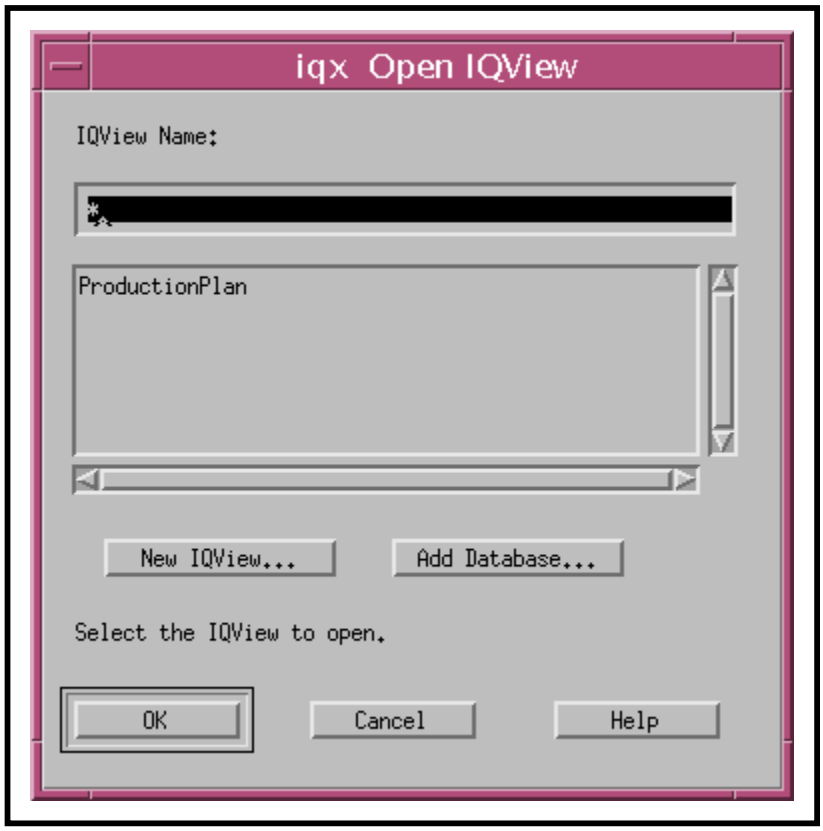


Figure 26.22.1-3. iqx Open IQView GUI

- If the **iqx License** dialogue box (Figure 26.22.1-4) is displayed, click on the **Cancel** button.

— The **iqx IQView** GUI (Figure 26.22.1-1) and either the **iqx Sybase Database Logon** GUI (Figure 26.22.1-2) or **iqx Open IQView** GUI (Figure 26.22.1-3) are displayed.

10 If the **iqx Sybase Database Logon** GUI (Figure 26.22.1-2) is displayed, go to Step 16.

11 If the **iqx Open IQView** GUI (Figure 26.22.1-3) is displayed and the desired IQView has been defined previously, perform Steps 12 through 14; otherwise, go to Step 15.

- If IQViews have been defined previously, they are listed on the **iqx Open IQView** GUI (Figure 26.22.1-3); otherwise, a list of database tables is displayed.

12 If the desired IQView has been defined previously, highlight the IQView to be opened by clicking on its entry in the list of IQViews.

13 Click on the **OK** button.

— The **iqx Sybase Database Logon** GUI (Figure 26.22.1-2) is displayed.

14 Go to Step 16.

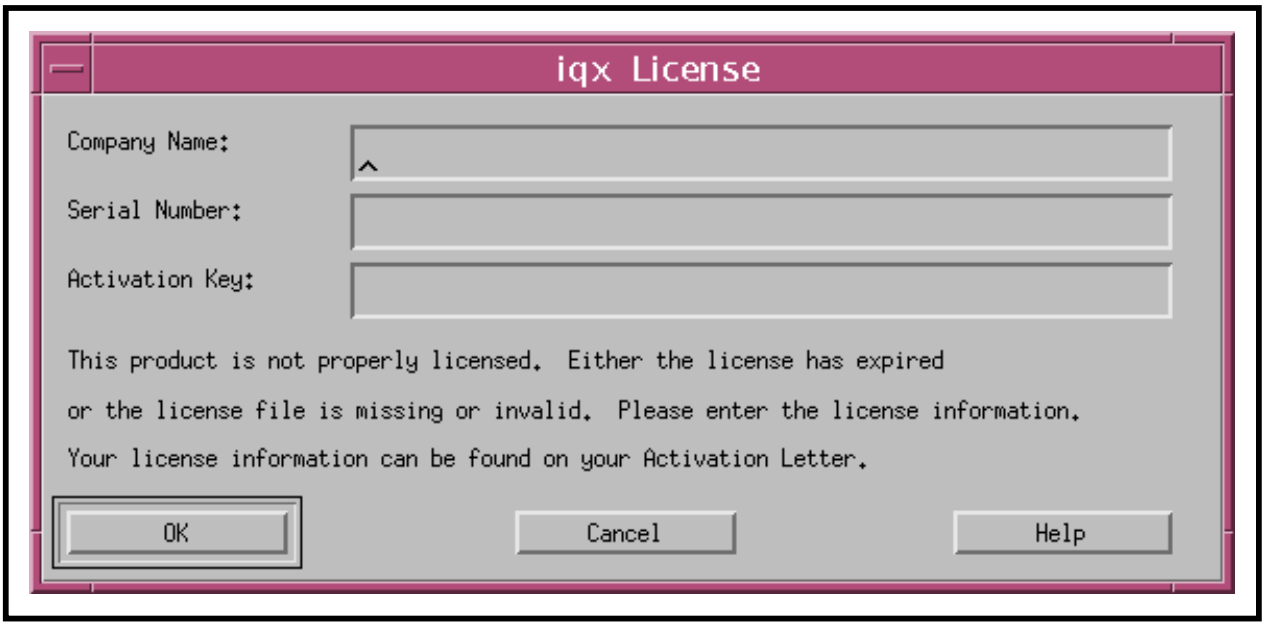


Figure 26.22.1-4. iqx License Dialogue Box

15 If the **iqx Open IQView** GUI (Figure 26.22.1-3) is displayed and the desired IQView has not been defined previously, click on the **Add Database…** button.

— The **iqx Sybase Database Logon** GUI (Figure 26.22.1-2) is displayed.

16 When the **iqx Sybase Database Logon** GUI (Figure 26.22.1-2) is displayed, type the appropriate entries in the following fields:

- **User:**

— For example: **pdpsUsr**

— The DAAC Database Administrator can provide the actual values to be entered.

— **Password:**

— For example: **dbpa$$wd**

— **Database:**

— For example: **pdps_TS1**

— **Server:**

— For example: **x0pls02_srvr**

— Click on the **OK** button.

— Either the **iqx Open IQView** GUI (Figure 26.22.1-3) or the **iqx IQView** GUI (Figure 26.22.1-1) is displayed.

— If the **iqx Open IQView** GUI (Figure 26.22.1-3) is displayed, continue with Step 19; if the **iqx IQView** GUI (Figure 26.22.1-1) is displayed, go to Step 22.

— Click on the **New IQView…** button.

— The **iqx Table Selection** GUI (Figure 26.22.1-5) is displayed.

— Move database table names between the **Available Tables:** and **Selected Tables:** lists as necessary by selecting (highlighting) the name of the table to be moved, then clicking on either the **Select** or **Remove** button (as applicable) to move the table name to the other list.
— Database tables and the columns within each table are described in the 311-series documents (e.g., 311-CD-503-001, Release 5A Planning and Data Processing Subsystem Database Design and Schema Specifications for the ECS Project). The documents are available on the ECS Data Handling System (i.e., at http://edhs1.gsfc.nasa.gov).

21 When the desired table(s) has/have been moved to the **Selected Tables:** list, click on the **OK** button.

- The **iqx IQView** GUI (Figure 26.22.1-1) is displayed.

22 Select **Output** → **Columnar** from the pull-down menu.

- The **iqx Columnar Output** GUI (Figure 26.22.1-6) is displayed.

23 Move database table column names between the **Available columns:** and **Selected columns:** lists as necessary by selecting (highlighting) the column to be moved, then clicking on either the **Select** or **Remove** button (as applicable) to move the column name to the other list.

- The order in which columns are listed in the **Selected columns:** list is the order in which the columns will be listed in the eventual report.
- Database tables and the columns within each table are described in the 311-series documents (e.g., 311-CD-503-001, Release 5A Planning and Data Processing Subsystem Database Design and Schema Specifications for the ECS Project). The documents are available on the ECS Data Handling System (i.e., at http://edhs1.gsfc.nasa.gov).

24 If changing the order in which columns are listed in the **Selected columns:** list, select (highlight) the column to be moved, then click on the **Up** or **Down** button as necessary to reposition the selected column.

12. Highlighted column changes position in the **Selected columns:** list.

25 When the desired columns have been moved to the **Selected columns:** list, click on the **OK** button.

- The **iqx IQView** GUI (Figure 26.22.1- 7) is displayed.

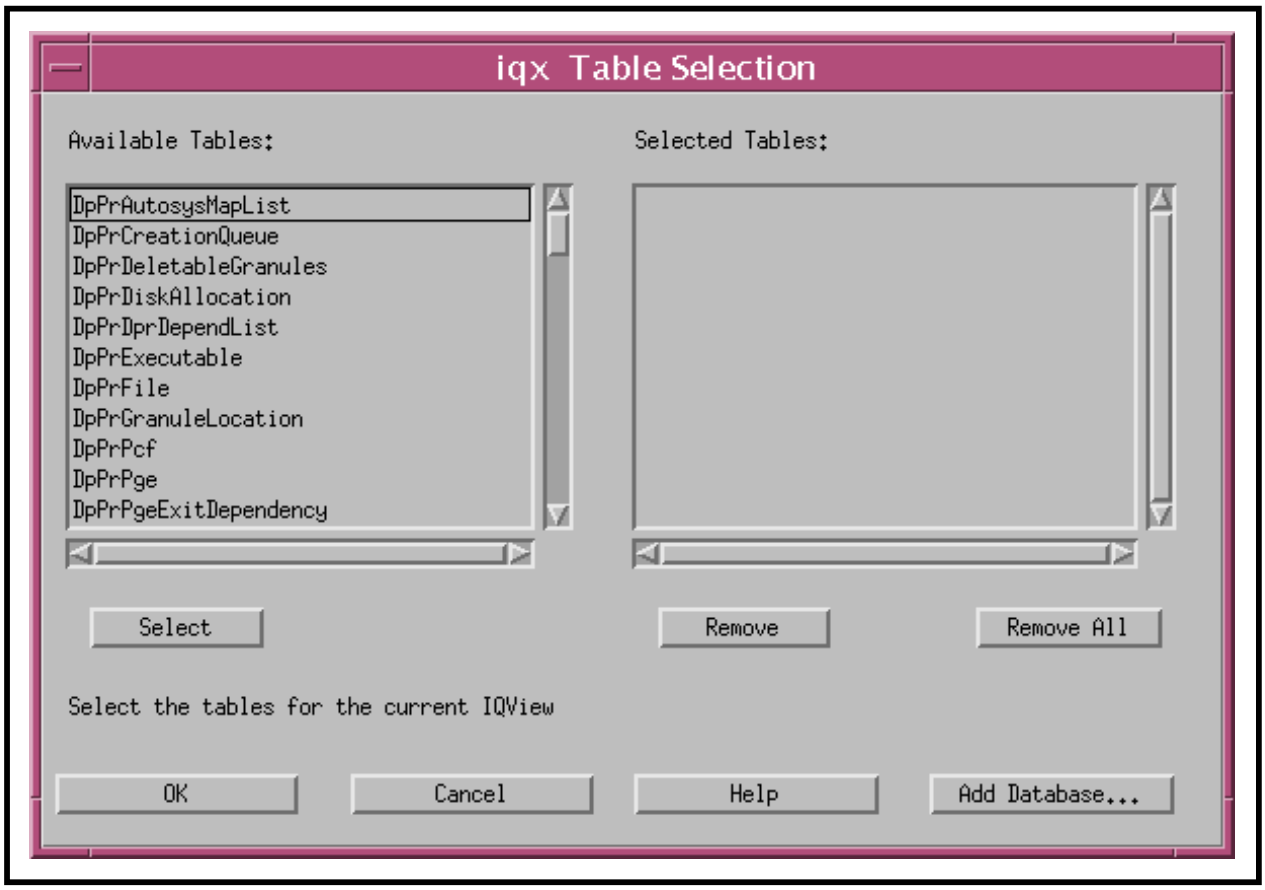


Figure 26.22.1-5. iqx Table Selection GUI

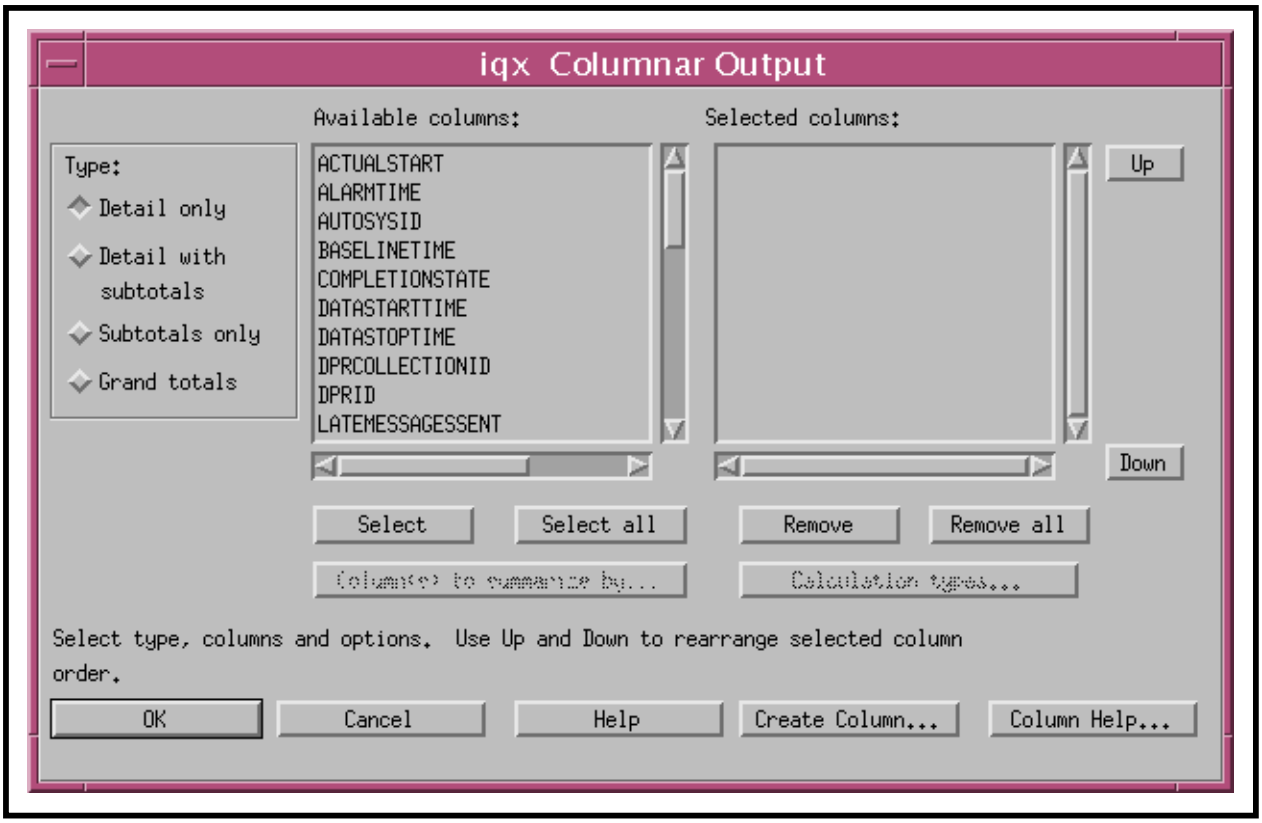


Figure 26.22.1-6. iqx Columnar Output GUI

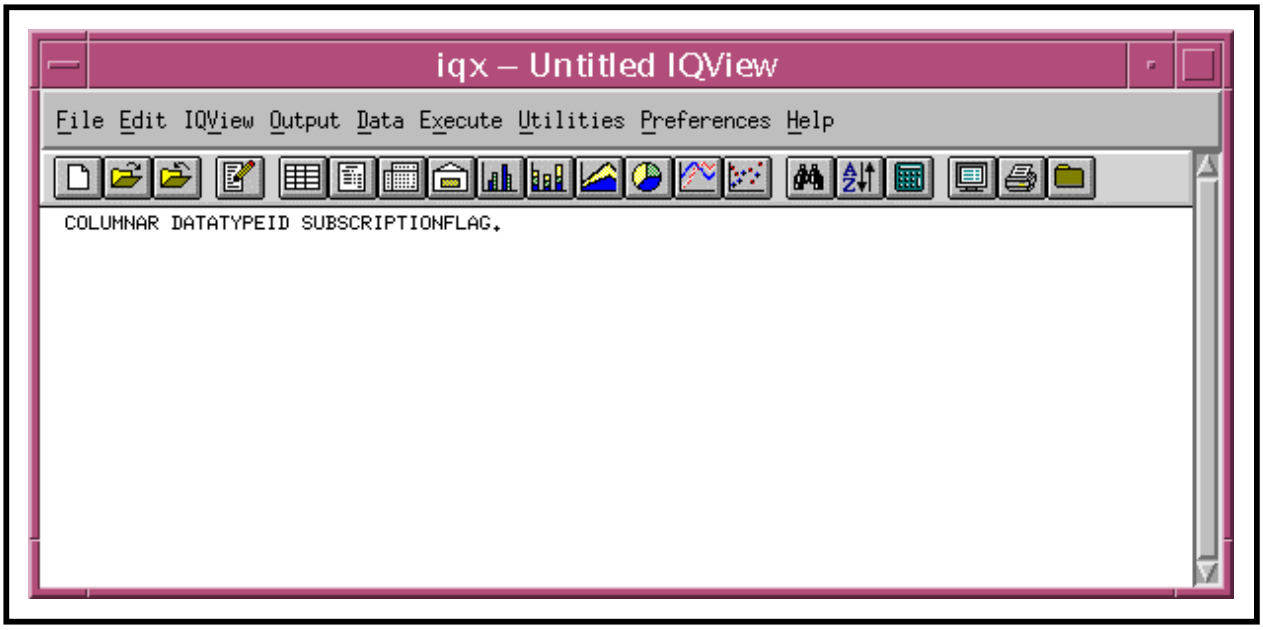


Figure 26.22.1-7. iqx IQView GUI

- The columnar selections are listed on the **iqx IQView** GUI as shown in Figure 26.22.1-7.

26 To generate a report make one of the following selections from the pull-down menu:

- **Execute → to Display** – to display the report on the terminal screen.
  - — The **iqx IQ Output** GUI (Figure 26.22.1-8) is displayed.
  - — Go to Step 35 after viewing the report.
- **Execute → to Printer** – to print the report.
  - — The **iqx Execute to Printer** GUI (Figure 26.22.1-9) is displayed.
  - — Go to Step 29.
- **Execute → to File** – to save the report in a file.
  - — The **iqx Execute to File** GUI (Figure 26.22.1-10) is displayed.
  - — Continue with Step 27.

27 Type a valid ***path/filename*** in the **Name:** field of the **iqx Execute to File** GUI (Figure 26.22.1-10).

- For example: **/home/cmshared/reportfile**
  - — Where **/home/cmshared/** represents the path and **reportfile** is the file name.

28 Click on the **OK** button.

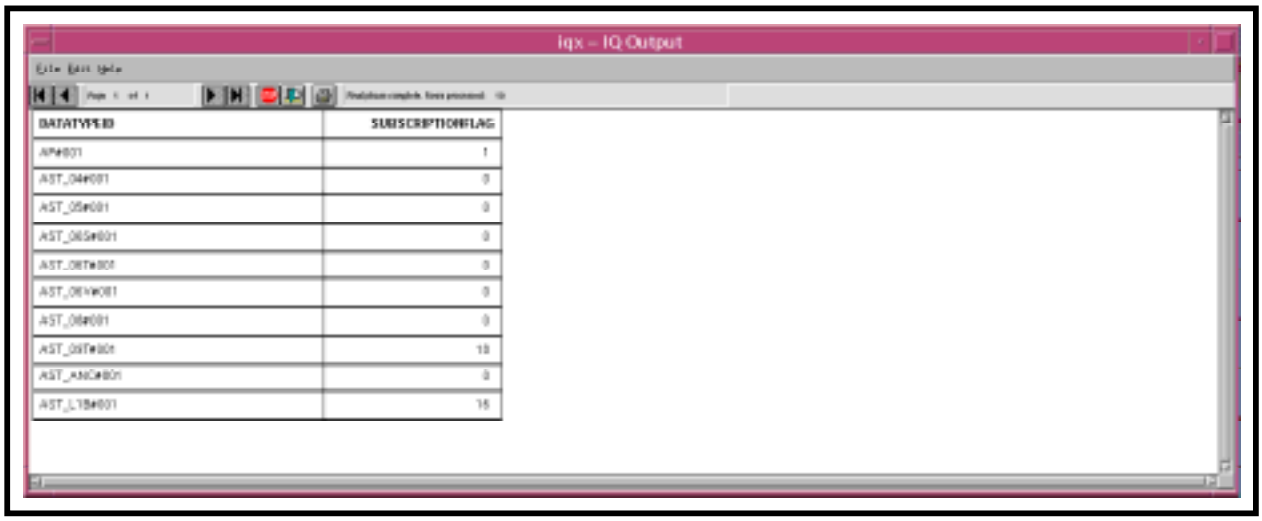


Figure 26.22.1-8. iqx IQ Output GUI

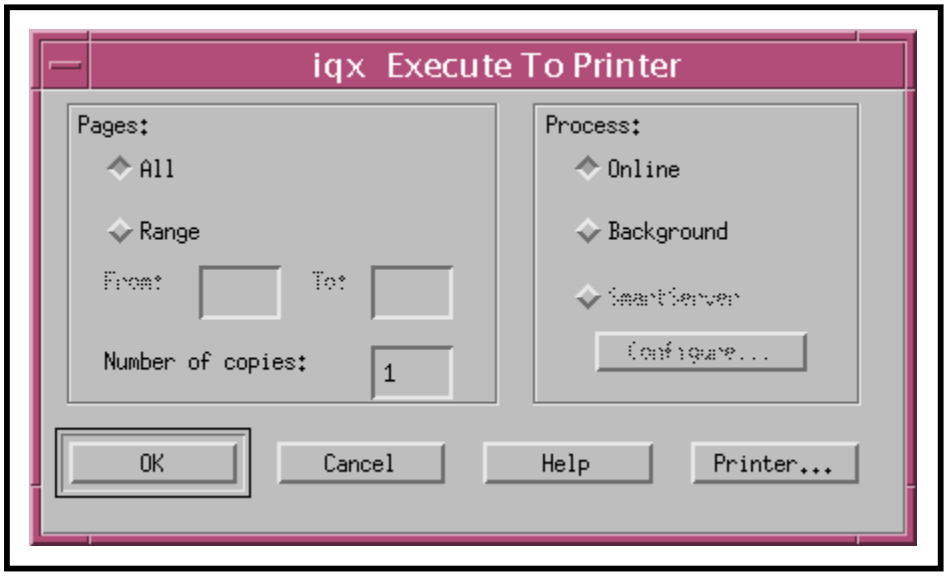


Figure 26.22.1-9. iqx Execute to Printer GUI

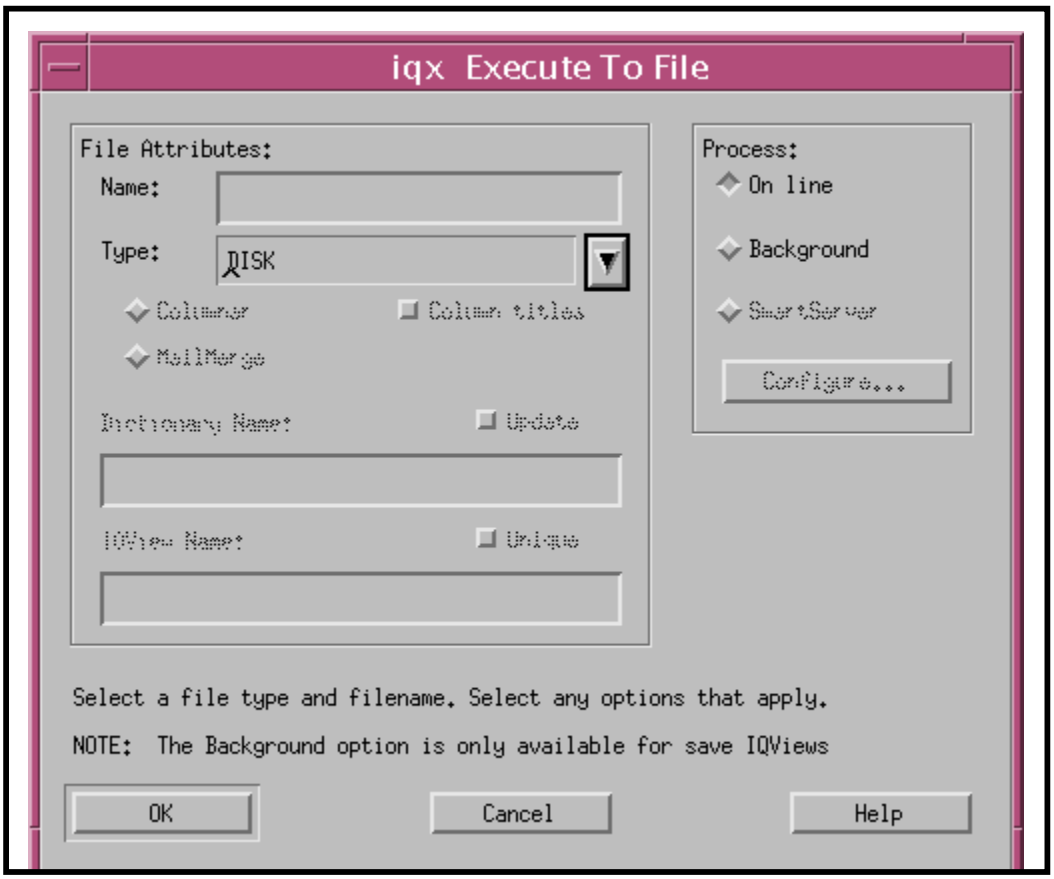


Figure 26.22.1-10. iqx Execute to File GUI

- Go to Step 35.

29 Click on the **Printer…** button on the **iqx Execute to Printer** GUI (Figure 26.22.1- 9).

- The **iqx Print Setup** GUI (Figure 26.22.1- 11) is displayed.

30 To list the available printers, first click on the option button associated with the **Printer** field.

- An option menu of printers is displayed.

31 Highlight the desired printer in the option menu.

- The desired printer is shown in the **Printer** field.
- For example: **Postscript printer one**.

32 If a report in landscape format is desired, click on the **Landscape** button.

33 Click on the **OK** button.

- The **iqx Print Setup** GUI (Figure 26.22.1- 11) is dismissed.

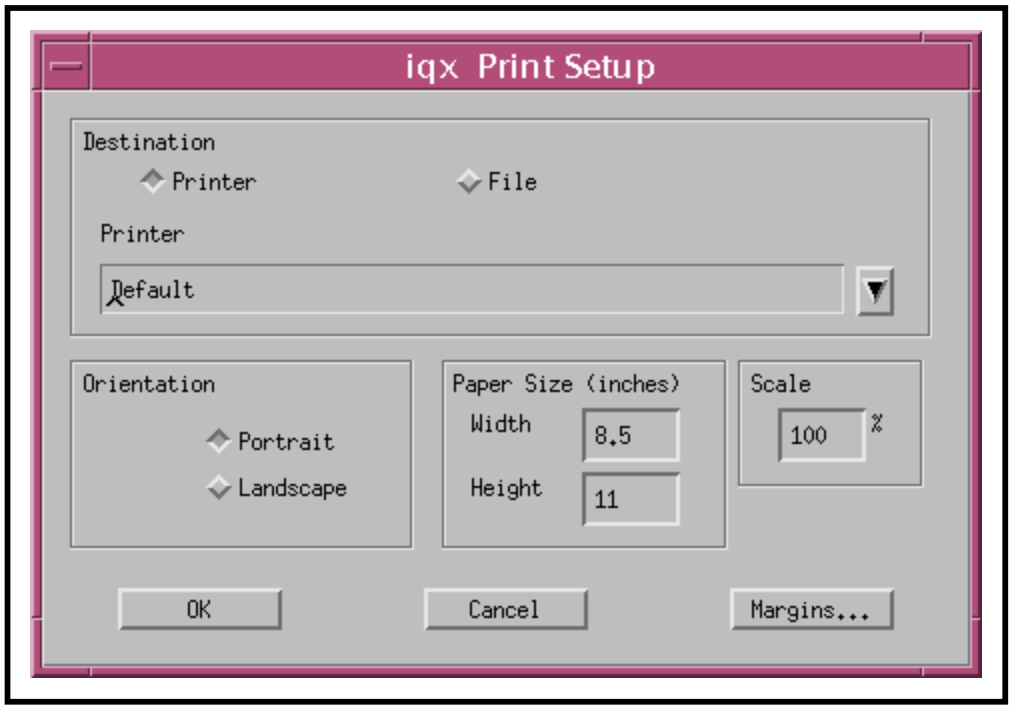


Figure 26.22.1-11. iqx Print Setup GUI

- The **iqx Execute to Printer** GUI (Figure 26.22.1-9) is displayed.

34 Click on the **OK** button.

35 To save the procedure/IQView, continue with Step 36; otherwise go to Step 43.

36 Select **File** → **Save Procedure As…** from the pull-down menu.

- The **iqx Save IQView** GUI (Figure 26.22.1- 12) is displayed.

37 Type a file name for the IQView in the name field.

38 Click on one of the following buttons if applicable:

- **Public**.
- **Public read only**.
- **Private**.

39 Click on the **OK** button.

- The **iqx Save Procedure** GUI (Figure 26.22.1-13) is displayed.

40 Type a file name for the procedure in the name field.

41 Click on one of the following buttons if applicable:

- **Public**.

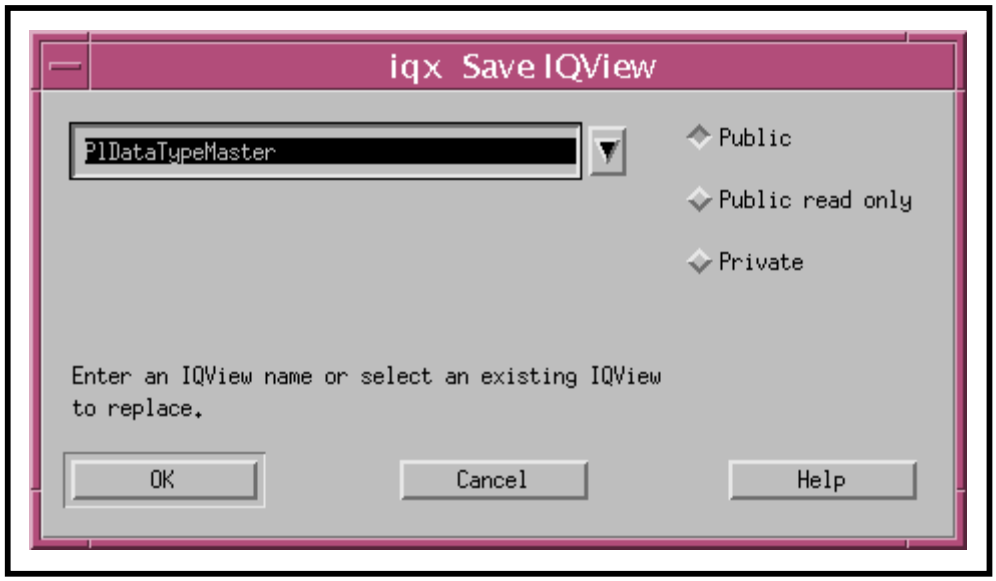


Figure 26.22.1-12. iqx Save IQView GUI

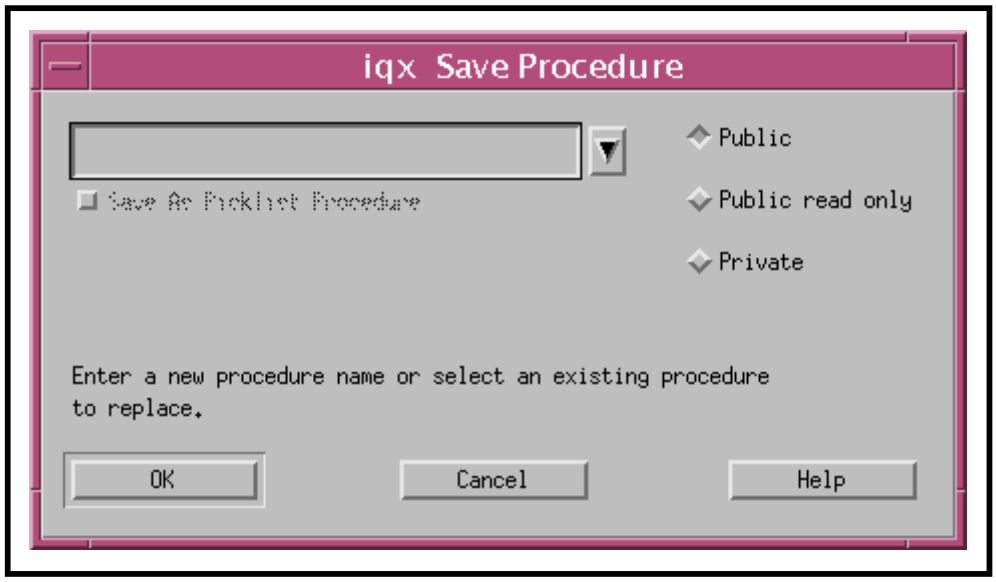


Figure 26.22.1-13. iqx Save Procedure GUI

- **Public read only**.
- **Private**.

42 Click on the **OK** button.

43 Select **File** → **Exit** from the pull-down menu to exit from the **iqx IQView** GUI (Figure 26.22.1- 1).

26.22.2 Formatting IQ Software Reports

[TBS]

This page intentionally left blank.