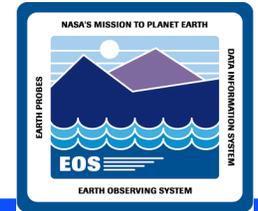


SCIENCE SOFTWARE INTEGRATION & TEST, Part 1 of 2

625-CD-616-001

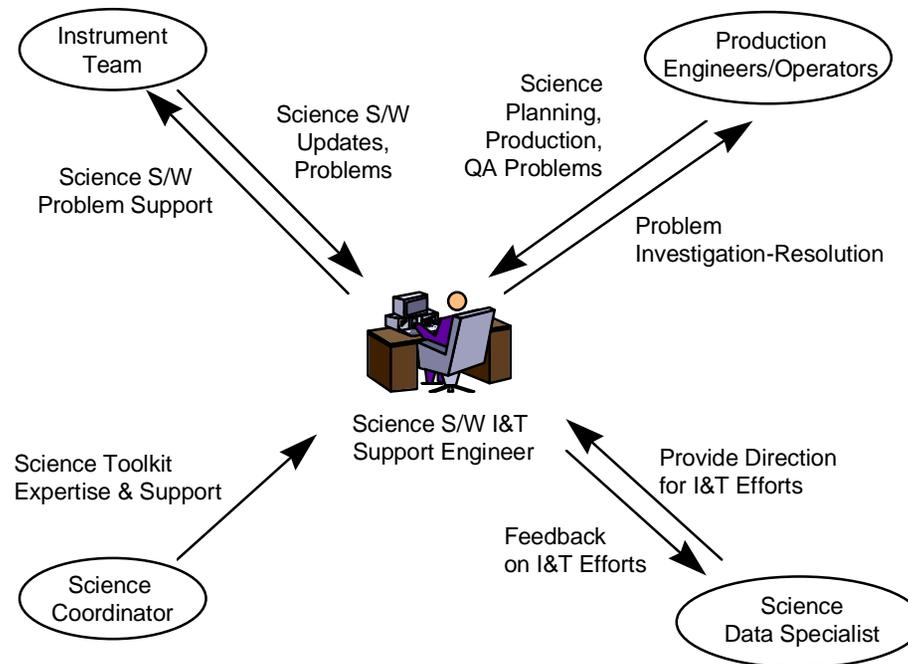
**ECS Version 2 Training
(updated for drop 6A)**

DAAC Science Software I&T Support Engineer Interfaces

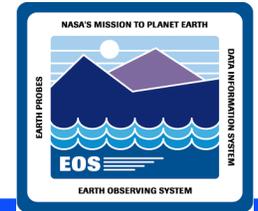


Provide DAAC SSI&T execution support, ECS tool and system expertise and science S/W processing problem support. Provide support to scientists in the development and integration of science software for both updates and new science software into the DAAC ECS system.

Interfaces

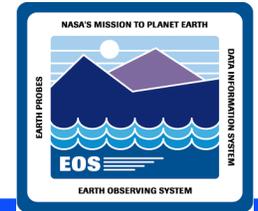


SSI&T Support Engineer Role and Responsibilities



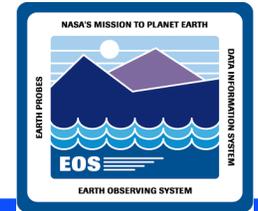
- 1. Provide support to Instrument Teams for the development and integration of science software into the DAAC ECS system.**
- 2. Perform standards checking on all delivered software, including source code, scripts, process control files and related documentation.**
- 3. Provide support for metadata updates and additions for science data products.**
- 4. Support science processing problem investigation and resolution.**
- 5. Recommend - assess, develop, implement changes to science toolkit software.**
- 6. Provide support for integration and test of new and modified toolkit functions into the science software.**

SSI&T Support Engineer Role and Responsibilities Continued



- 7. Support and track configuration of delta changes to science software, calibration coefficient files, relevant input files and documentation.**
- 8. Facilitate migration and integration of major and minor modified versions of the science software into production versions.**
- 9. Assess impacts and support integration and test for production planning and processing software changes. (Planning Workbench GUI)**
- 10. Provide feedback and receive direction on I&T efforts.**
- 11. Recommend-assess improvements for automated tools for SSI&T activities, such as file comparison and viewing data.**
- 12. Write and implement procedures to examine non-standard auxiliary files and files not in HDF EOS format.**

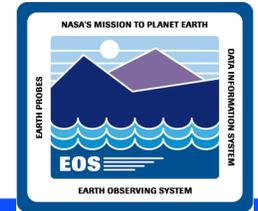
Science Software Integration and Test (SSI&T) Overview



WHAT ECS SSI&T IS

- **SSI&T is the process by which science software developed by Instrument Teams at local SCFs is tested and integrated into the ECS at the DAACs.**
- **The scope of SSI&T for pre-launch releases covers activities starting with delivery of the science software to the DAACs and ending with either the successful integration of each delivered PGE into ECS or the scheduled end date for SSI&T support.**
- **SSI&T is a team effort which can only be successful in the allotted time if all groups cooperate.**

Science Software Integration and Test Overview

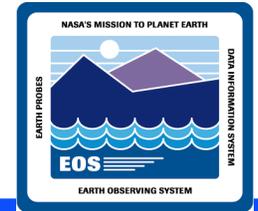


WHAT ECS SSI&T IS NOT

- **SSI&T is not validation of science algorithms that are incorporated into PGEs which produce science data products.**
- **SSI&T is not validation of the science data produced.**
- **Although the Operational Procedures are written as checklists of menu driven activities, SSI&T is not a turnkey process which can be run by test personnel who have no knowledge and experience related to science software development and data processing.**
- **SSI&T is not a simulation of production.**

SSI&T Training Objectives

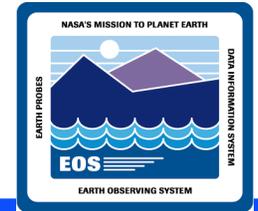
Day 1



- **Overall: Provide the proficiency to integrate and test science software into EOSDIS**
 - Release 4 Architecture Overview
 - The ECS Assistant/Scripts/Whazzup
 - Preparation and Setup - alias file
 - Review of Science Office Project Instructions (at: http://dmserver.gsfc.nasa.gov/proj_instr/sopi_index.html)
 - Acquiring and Unpacking Delivered Algorithm Package
 - Performing a DAP Acquire Using SSIT Manager
 - Science Software Configuration Management
 - The SSIT Manager
 - Standards Checking of Science Software
 - Compiling and Linking Science Software
 - Running a PGE in a Simulated SCF Environment

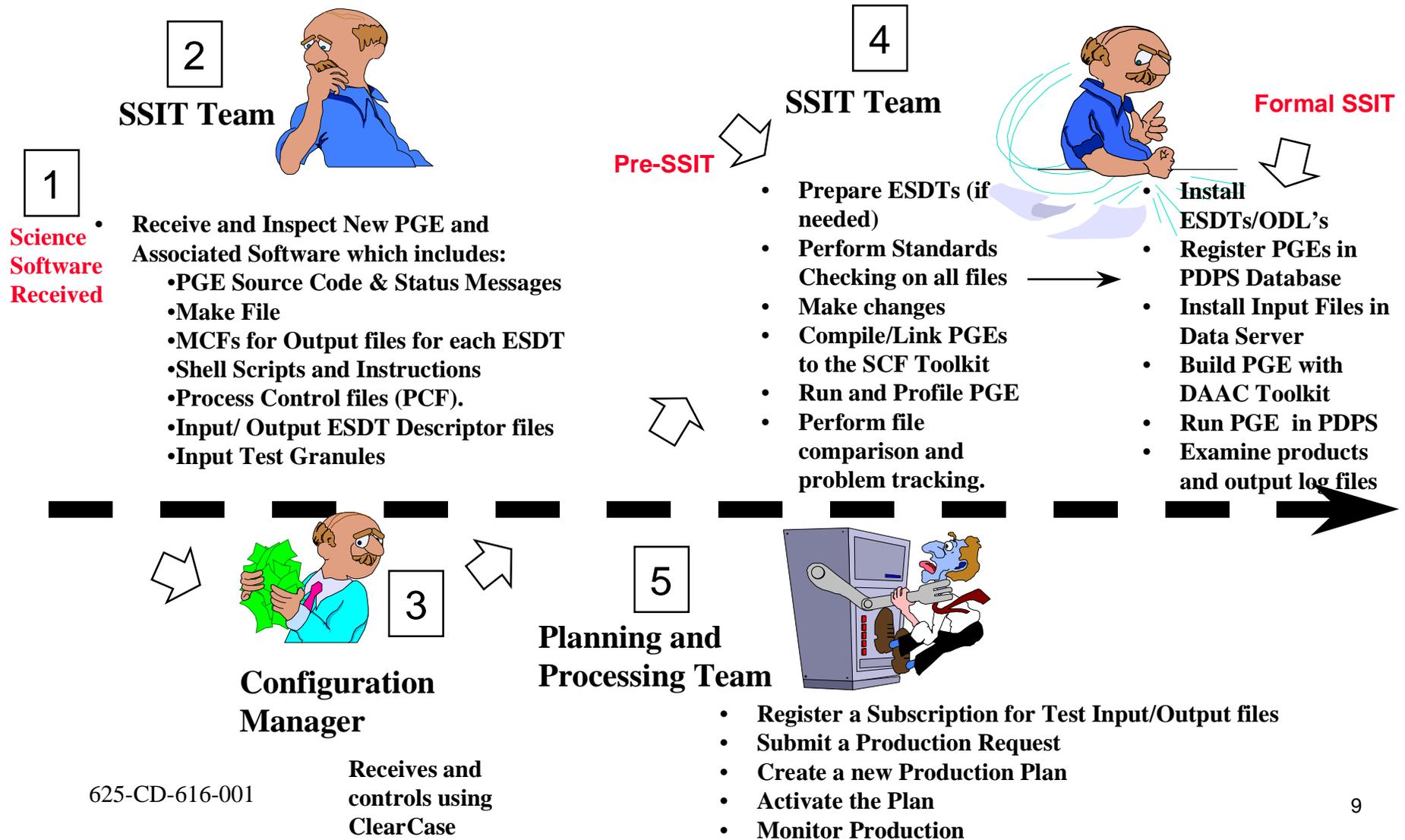
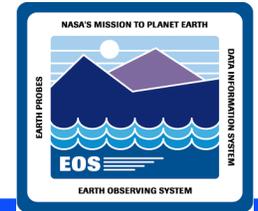
SSI&T Training Objectives

Day 2

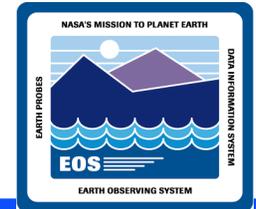


- Refer to the Post Processing and General Investigation Section for :
 - Examining PGE Produced Log Files &
 - File Comparison and Data Visualization
- Preparation of Earth Science Data Types (ESDTs/DLLs)
- ODL Preparation
- Production Rules
- DPREP
- Update the PDPS Database and Data Servers
- PGE Planning and Processing
- Post Processing and General Investigation

SSI&T Process Overview - The Big Picture



Architecture Overview



Release 4 Architecture: Overview

ECS Subsystems

Client Subsystem (CLS)

Interoperability Subsystem “ADSRV” (IOS)

Data Management Subsystem (DMS)

Data Server Subsystem (DSS)

Ingest Subsystem “STMGT” (INS)

Data Processing Subsystem (DPS)

Planning Subsystem (PLS)

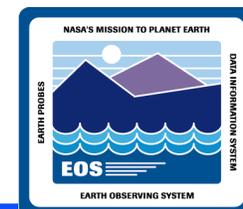
Communications Subsystem (CSS)

Management Subsystem (MSS)

Internetworking Subsystem (ISS)

Implications for SSI&T Procedural differences

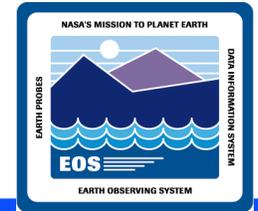
Major SSI&T Functions V2



| Function | | Release VERSION 2.0 |
|--------------------------------|--|--|
| System Operation | | All servers must run and communicate with each other; bring up manually, or use ECSAsisst 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. 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/Granule Deletion |
| Subscription Management | | Subscription Manager |

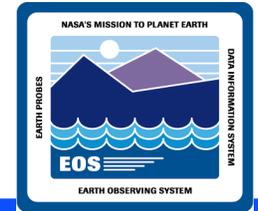
Table 1. ^{625-CD-616-001} Major SSI&T Procedures for Version 2.

Release 6A Capabilities



The Earth Observing System (EOS) Data and Information System (EOSDIS) Core System (ECS) capabilities are developed in terms of formal releases. Release 6A, which is controlled by Configuration Management, provides capabilities to support the ingest and archive of raw data obtained from the EOS AM 1 mission spacecraft, morning equator crossing spacecraft series (Terra (AM-1)), EOS PM 1 mission spacecraft, afternoon equator crossing spacecraft series (Aqua (PM-1)) and the Land Remote-Sensing Satellite (Landsat 7). Other capabilities provided by Release 6A include processing the data obtained, distributing raw or processed data as requested, quality assurance of processed data, supporting communication networks, and systems monitoring via interfaces with the ECS operations staff.

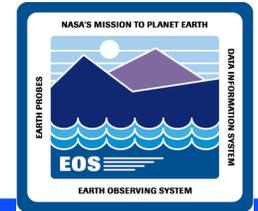
Release 6A unique capabilities Continued



Release 6A unique capabilities include:

- support the regeneration and archive of products previously produced and archived
- support ingest, production, storage, and distribution of Terra and Aqua instrument data
- support writing files to CD-ROMs and DLT tape drives for distribution
- support data compression (Unix and Gzip) for distribution
- provide access to non-science data collections by a limited number of attributes and values

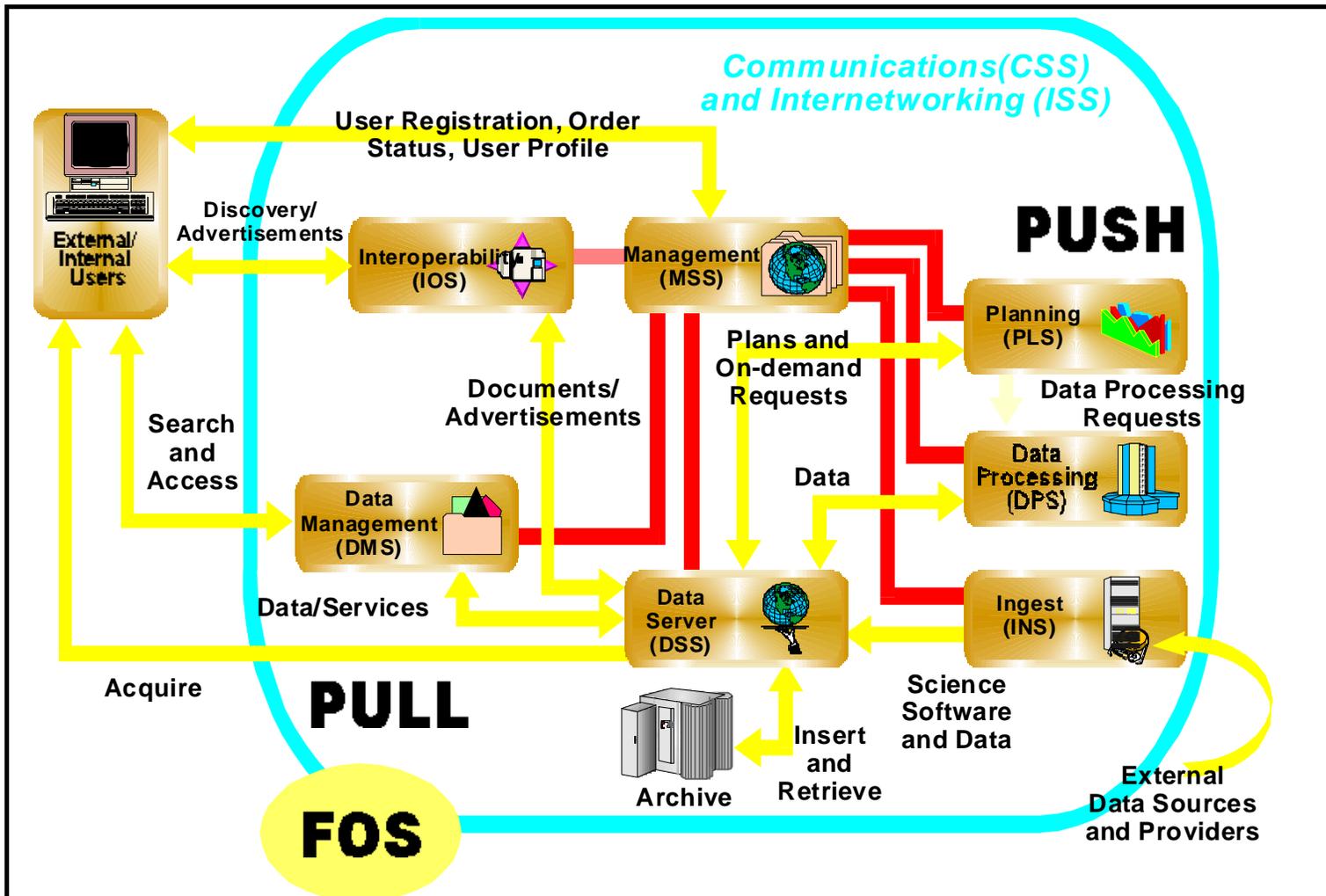
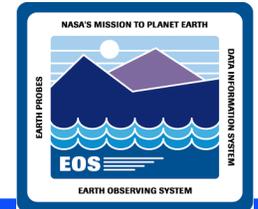
Release 6A unique capabilities Continued



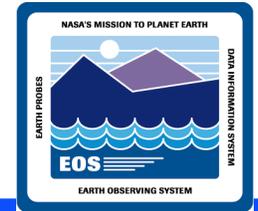
- provide Science Data Server scalability/operability improvements
- support FTP Pull Subscription
- provide machine-to-machine gateway support between SIPS and ECS for data orders
- provide capability for operator deletion of granules
- support EOS Data and Operations System (EDOS) backup by distribution of replacement data to EDOS on D3 tapes and ingesting of backup data on D3 tapes from EDOS
- archive and distribution request management by multi-host scheduling
- ingest browse and metadata from IGS tapes
- allow users to request a data processing request associated with a DAR at the time of submittal

A more detailed overview of the Release 6A ECS may be found in the Release 6A Segment/Design Specifications for the ECS Project, 305-CD-600-001.

Communications and Internetworking



The ECS Assistant



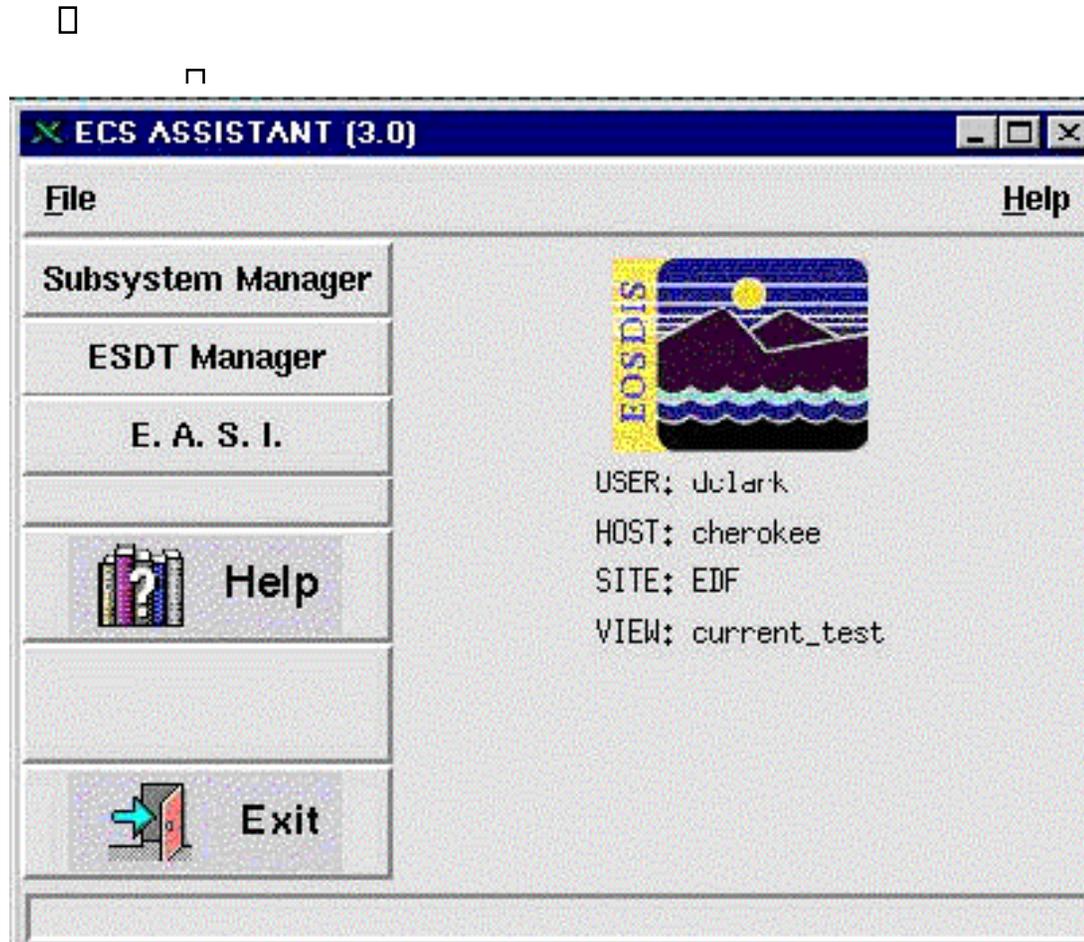
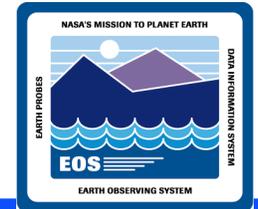
Using ECS to Perform Sub System Management Functions

Using ECS Assistant to Stage ESDT's

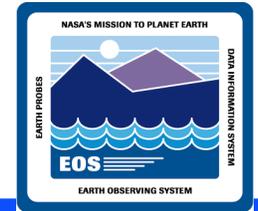
Using ECS Assistant to view Data Bases

Using ECS Assistant to Perform Installs

ECS Assitant GUI



Subsystem Manager GUI



ECS ASSISTANT: Subsystem Manager User: cmts1 Host: p0acs03 Site: PVC

File Tools Options Latest Info Extensions Help

Common Tasks cleanup database install kill mkcdsentry mkcfg monitor package stageinstall start

viewlog uninstall

| Modes | Subsystems | Components | Applications | Executables |
|------------------------------------|--|--|-----------------------|---------------------------------------|
| OPS SHARED TS1 TS2 TS3 | CLS CSS DM DPS INGEST IDS MSS PLS TODLKIT VOC | EcDsSr (sdsrv) EcDsSt (stmgmt) EcDsDd (distribution) EcDsDo (ddsrv) | EcDsScienceDataServer | EcDsScienceDataServer EcDsSdSrvGui |

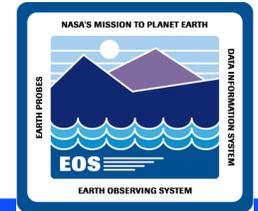
New Refresh

Results

| Settings | Installation Statistics |
|---|--|
| clearcase view : <NULL> Mode : <TS1> Subsystem : <DSS> Source File Loc. : <NULL> Installation Type : <NULL> | Install Record Entry Installed by : uid=5088(cmts1) gid=80(cmts1) Date/Time : 05/03/2000 13:38:06 Clearcase view : NONE Stage Location : MODE Staging Mode : /net/p0msh11/codedrop/5B.03 PVC/SUN Installation Type : NONDEV Subsystem installed : DSS Component installed : EcDsSr |

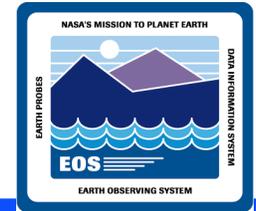
Perfect Installation
 Install Warnings
 Install Errors

The ECS Assistant Functionality Replaced in Part by Scripts and Monitor GUI Whazzup



Currently, the ECS Assistant has lost its capability to monitor ECS. This capability now exist with GUI Whazzup. ECS Assistant continues to have the ability to Monitor Subsystem functions and is mainly used for doing ECS Assistant Subsystem Installs (E.A.S.I.) and staging ESDT/DLL's into the directories, CUSTOM/data/ESS and CUSTOM/lib/ESS respectively. Database Review capability still exists however. The ESDT Manager Installation/Deletion functions are no longer available, the ability to startup and shutdown subsystem servers and the use of the ECS Logfile Viewer have been taken away in Release 6A.

Scripts and Monitor GUI Whazzup/ISQL Browsers



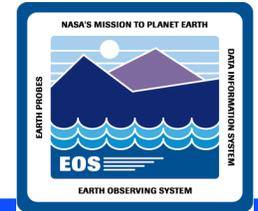
The use of scripts provides users with the means to perform functions such as subsystem server startup and shutdown. During the course of performing these tasks, SSI&T operators can use the following Scripts to perform the following functions:

- **To start up or shut down all servers at the same time**, a **Script** is used that accesses a list of subsystem servers.
- **To start up and shut down servers individually** using a **Script** established within each subsystem.
- **To graphically monitor the server up/down status** with the **Whazzup GUI**
- To view ESDTs for SSI&T
- To **review various databases** used in the ECS system by using an **ISQL Browsers** established in each subsystem.

In the following sections, we will address aspects of how to use **Scripts, the Whazzup GUI** and portions of ECS Assistant in our SSI&T activities.

- **Section one** explains how to use **Scripts** to facilitate and manage the subsystems and their servers, including **server start up and shut down**.
- **Section two** contains an **ESDT monitor function** which includes reviewing the Science Data Server database through the **DB Viewer GUI** provided by ECS Assistant

Using Scripts to Start Up / Shut Down Servers



The DAAC's may have established their own scripts to Start Up/Shut Down Subsystem servers. This procedure describes Scripts that are used at Landover on the VATC and PVC systems to start up and shut down subsystem servers. The procedure described here will apply to all the servers from different subsystems as well as individual servers. Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow

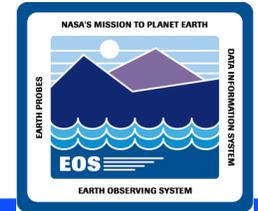
All Servers **Start Up/Shut Down**

- 1 Script name: **rcmd_start_mode TS1, rcmd_kill_mode TS1**
- 2 Location: **/home/cmshared/bin/**
- 3 **HOSTLST** contains all the Subsystem Servers that will be operated upon.

Individual Servers **Start Up/Shut Down**

- 1 If you want to Start or Kill a server, you must be on the individual machine that supports the server in question.
- 2 Type **cd /home/cmshared/bin/ <ENTER>**
- 3 Type **start_mode TS1** to start up a server or **kill_mode TS1** to shutdown a server.
<ENTER>

Monitoring ECS using WHAZZUP Web GUI



To use the **WHAZZUP GUI** to monitor ECS make the following entries to display the GUI.

Log into a server that has Web access.

Setenv DISPLAY ..:0.0

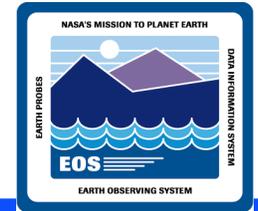
/tools/bin/ssh -l cmshared t1code1u.ecs.nasa.gov

Password:

setenv DISPLAY :0.0

ssh t1pls02

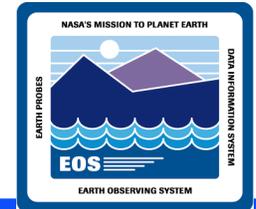
WHAZZUP Continued



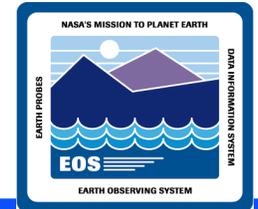
Enter `/tools/bin/netscape/` when the Netscape GUI is displayed, then select from the Bookmark: **whazzup**. A GUI such as the one depicted in the slide above should appear on the terminal screen.

Notice the other links listed in the bookmark List depicted in next slide that go to sites that support ECS.

Server Monitor WHAZZUP

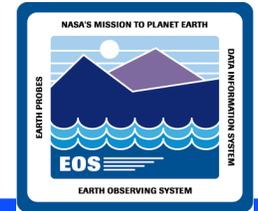


WHAZZUP Bookmark List



- Add Bookmark Alt+K
- File Bookmark
- Edit Bookmarks... Alt+B
- ✂ The Java DAR Tool on t1ins01 in OPS mode
- ✂ Netscape Search Re... ECS, graph master
- ✂ Big Brother - Netw... 12:06:49 EDT 1998
- ✂ EOSDIS User Registration
- ✂ Trouble Ticketing: t1msh01
- ✂ RELB I&T HOMEPAGE
- ✂ Goddard Space Flig...e Directory (ESOD)
- ✂ Trouble Ticketing: Menu
- ✂ EOSDIS Version 0 W...ts for ECS Testing
- ✂ \$Site's Earth Scie...e Directory (ESOD)
- ✂ EOSDIS Home Page
- ✂ EOS Data Gateway at ly:a.gsfc.nasa.gov
- 📁 Personal Toolbar Folder
- ✂ EOS Data Gateway at ly:a.gsfc.nasa.gov
- ✂ EOSDIS Home Page
- ✂ Netscape Server Administration
- ✂ Netscape Server Administration
- ✂ Lotus cc:Mail Web
- ✂ 401 Authorization Required
- ✂ EOS Data Gateway at ly:a.gsfc.nasa.gov
- ✂ Whazzup?? v2.0
- ✂ EOS Data Gateway at ly:a.gsfc.nasa.gov
- ✂ ECS Pete Server
- ✂ EDHS Homepage
- ✂ SMC - Main Page

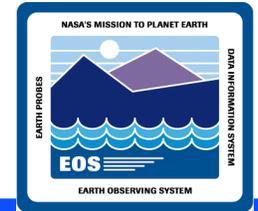
Preparation and Setup



Preparation for SSI&T is a cooperative effort by M&O and SSI&T

- **M&O Activities for Preparation and Setup:**
 - Setup of user accounts.
 - Setup ClearCase VOB.
 - Installation of Software Drops
- **SSIT Preparation and Setup:**
 - Examine .cshrc file.
 - Set additional environmental variables.
 - Source additional setup files.
 - Verify access to ClearCase VOB.
 - The Green Book SSI&T Operational Procedures provides a road map to get from a Delivered Algorithm Package to Science Software which is integrated into ECS and ready for production.
 - Procedures are ordered in a logical sequence.
 - Some deviation will be required for actual work at each DAAC.

Science Software Configuration Management



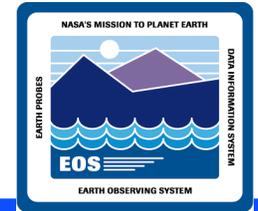
ClearCase - COTS tool for configuration management of science software.

Invocation Methods - Command line or Graphical User Interface (GUI).

Key Terms -

- **Versioned Object Base (VOB) - a mountable file system which stores version controlled data in directories and files.**
 - **Any Unix file : source files, script files, documents, spreadsheets.**
 - **Binary data and object files are not stored efficiently.**
 - **Usually accessed with standard UNIX and ClearCase Tools.**
- **View - A working context for a user. Used to access any VOB to make files and directories visible and accessible. Comprised of a storage area for checked out files.**
- **Element - File or directory in ClearCase VOB.**

Creating and Setting a View



Naming Conventions

- Provides file/directory names for locating directories or files.
- Key Names
 - ViewName - name of the user's view.
 - PathName - pathname is the path to the VOB directory.

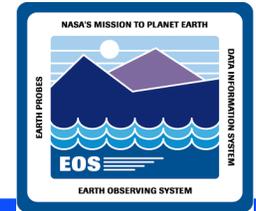
Scope

- Needs to be created only once.
- Must be set at beginning of each user session.

Key Assumptions

- ClearCase is available.
- A VOB has been created.

Creating a View

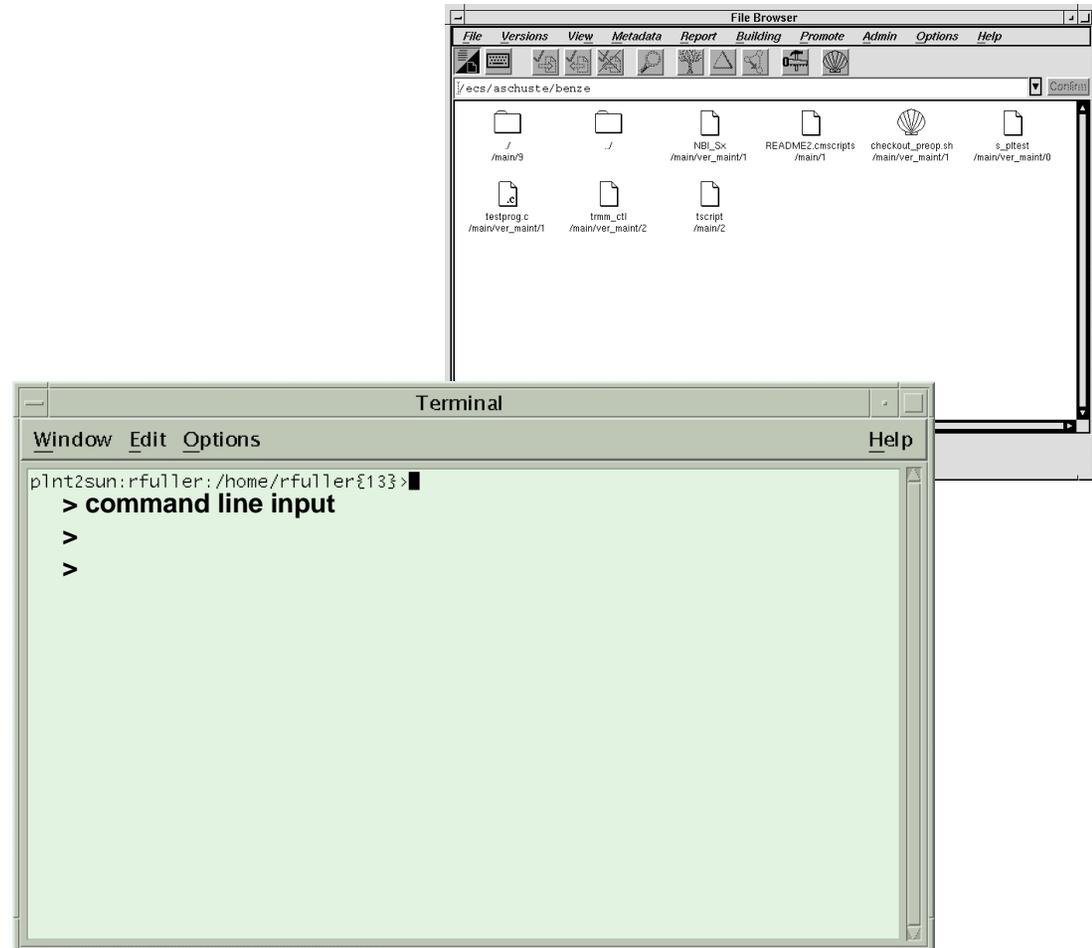


Access ClearCase by typing:
cleartool

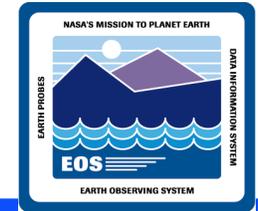
Create a view by typing:
mkview -tag *ViewName*
ViewPath/ViewName.vws

View is created named
ViewName

ViewName = name of view
ViewPath = path to view directory
vws = file extension



Setting a View

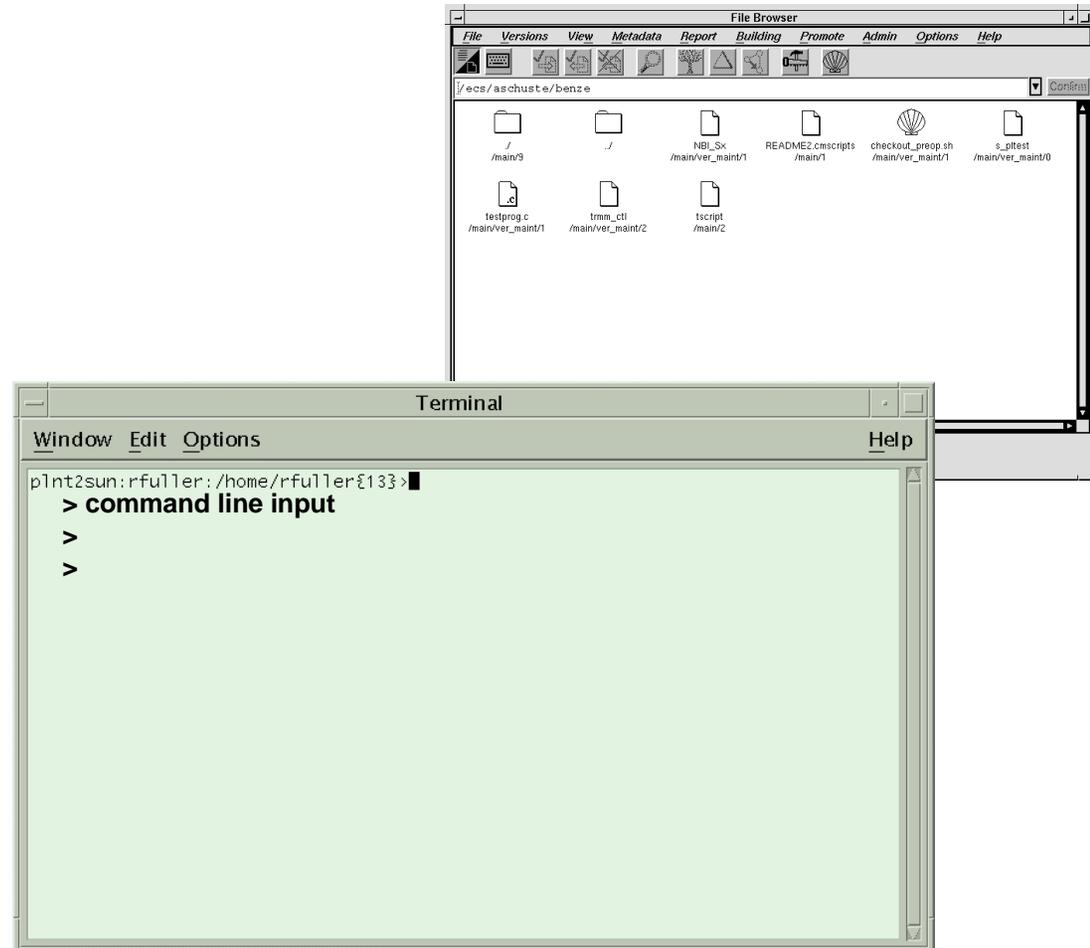


Access ClearCase by typing:
cleartool

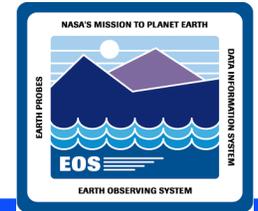
Set a view by typing:
setview *ViewName*

View is set to *ViewName*

ViewName = name of view



Entering a New Directory into ClearCase



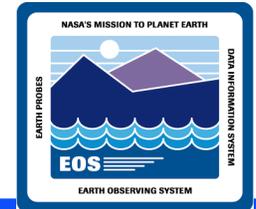
Key Assumptions

- A VOB and subdirectory has been created to hold the file
- A view has been created.

ClearCase Commands

- `cleartool setview ViewName` - Launches ClearCase and displays the user's view.
- `cd PathName` - Changes directory to a subdirectory in the VOB.
- `cleartool checkout -nc .` - Checks out a directory from ClearCase.
- `cleartool mkdir -nc DirName` - Creates a new directory (subdirectory).
- `cleartool checkin -nc DirName` - Checks new directory into ClearCase.
- `cleartool checkin -nc .` - Checks the current directory into ClearCase.

Creating a Directory



Access ClearCase by typing:
cleartool setview *ViewName*

Change Directories by typing:
cd *PathName*

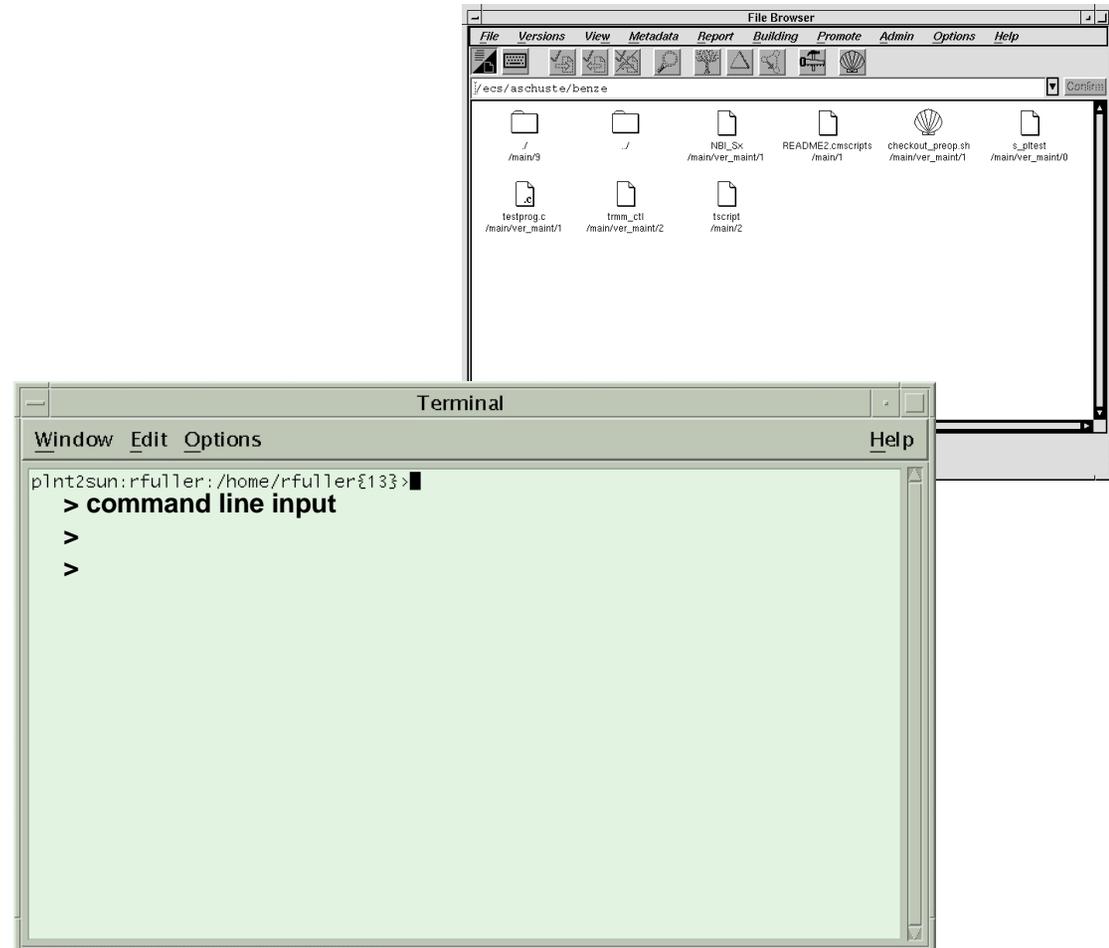
Checkout directory by typing:
cleartool checkout -nc .

Create a new directory by typing:
cleartool mkdir -nc *DirName*

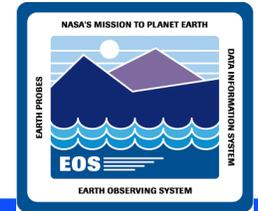
Checkin a directory by typing:
cleartool checkin -nc *DirName*

A new directory has been created and checked into ClearCase.

625-CD-616-001



Entering a Single File into ClearCase



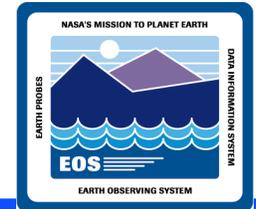
Key Assumptions

- A VOB and subdirectory has been created to hold the file
- A view has been created.

ClearCase Commands

- `cleartool setview ViewName` - Launches ClearCase and displays the user's view.
- `cd PathName` - Changes directory to a subdirectory in the VOB.
- `cp FilePath/FileName .` - Copies file from current to VOB directory.
- `cleartool checkout -nc .` - Checks out the current directory.
- `cleartool mkelem -nc FileName` - Creates a new element/file.
- `cleartool checkin -nc FileName` - Checks the file into ClearCase.
- `cleartool checkin -nc .` - Checks the current directory into ClearCase.

Entering a Single File into ClearCase



Access ClearCase by typing:
cleartool setview *ViewName*

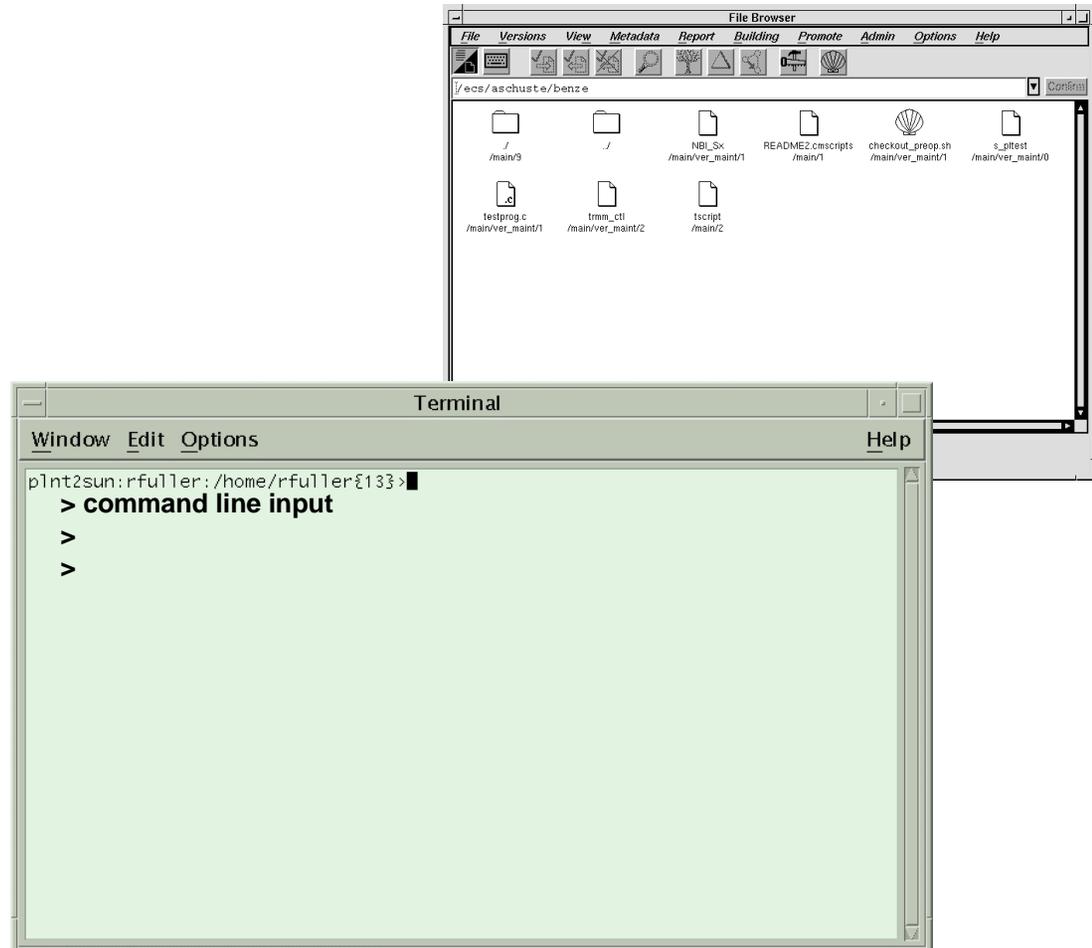
Change Directories by typing:
cd *Pathname*

Checkout directory by typing:
cleartool checkout -nc .

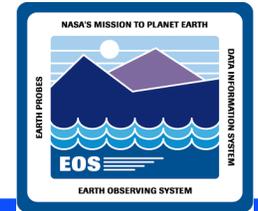
Create a new element by typing:
cleartool mkelem -nc *FileName*

Checkin a file by typing:
cleartool checkin -nc *FileName*

Checkin a directory by typing:
cleartool checkin -nc .



Importing Files into ClearCase



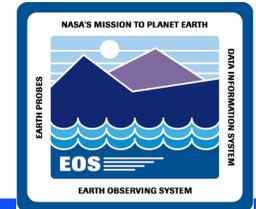
Key Assumptions

- DAAC SA required to complete this procedure.
- A VOB and subdirectory has been created to hold these files.
- No object files or executables exist in the directory.
- The PGE was received with a directory structure that contains various types of files.
- The PGE directory structure will be maintained.

ClearCase Commands

- `cd ParentDir` - Changes directory to the parent directory of the directory structure to be brought into ClearCase.
- `clearcvt-unix -r DirName` - Creates a conversion script to import everything in *DirName* directory and everything below it to ClearCase.
- `cvt_script` - Name of created script and command to run the script to place all elements under ClearCase.

Importing Multiple Files into ClearCase



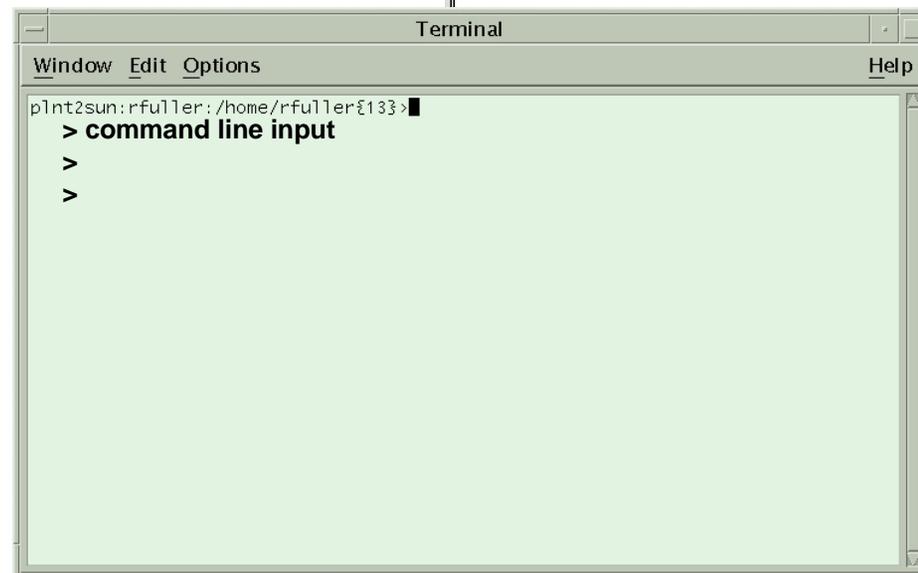
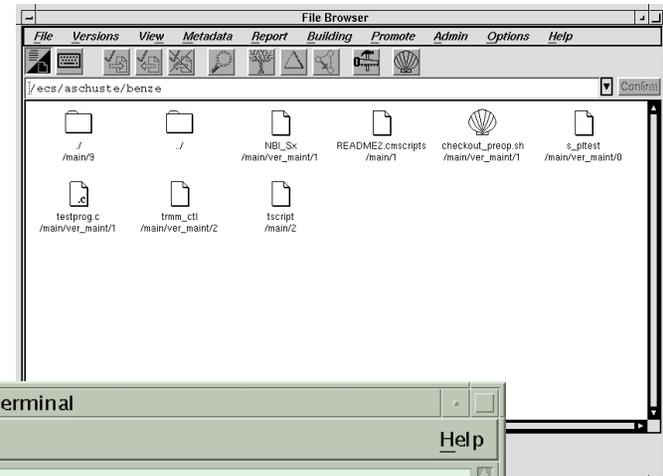
Change directories to parent of directory structure:
cd *ParentDir*

Create a conversion script :
clearcvt-unix -r *DirName*

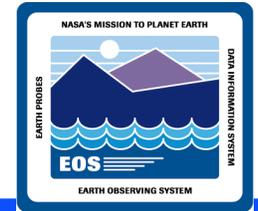
The output script is named:
cvt_script

Change directories in VOB by typing:
cd *Pathname*

Copy files into ClearCase by typing the script: **cvt_script**



Checking Out an Element from ClearCase



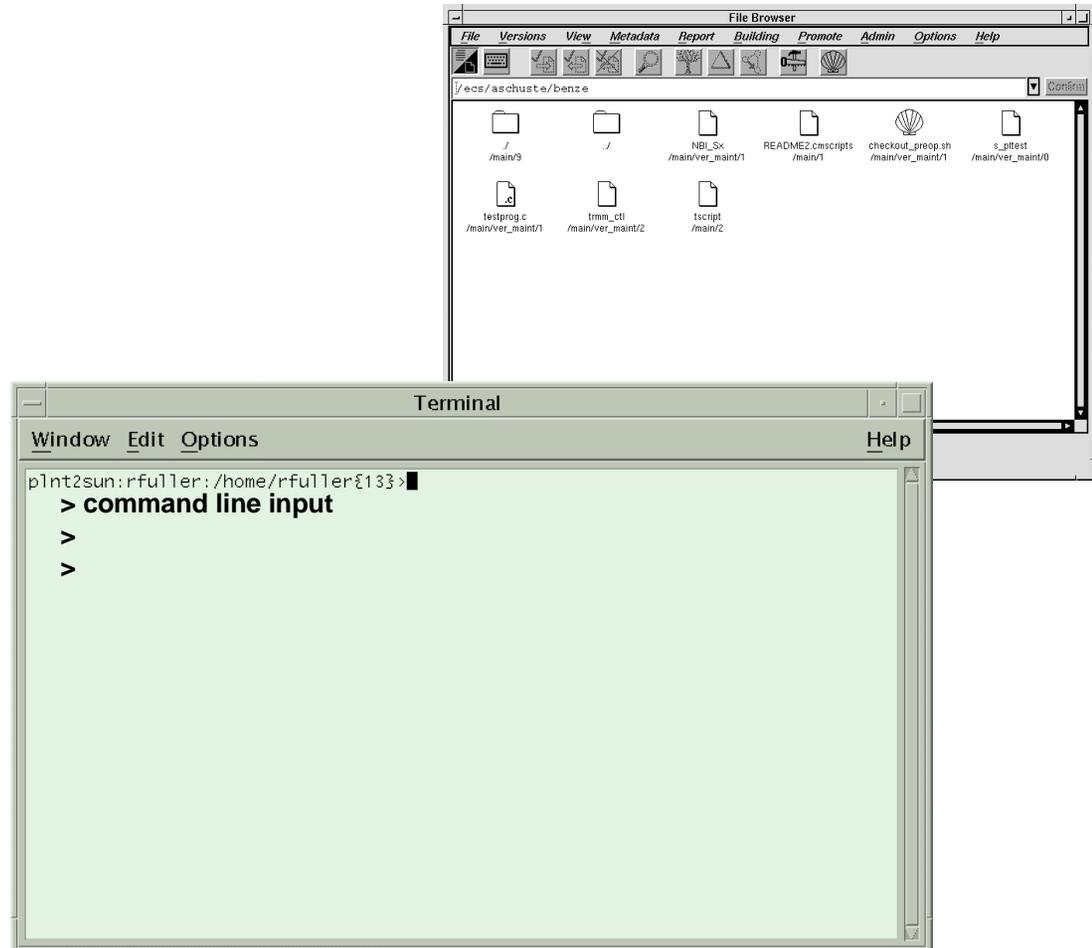
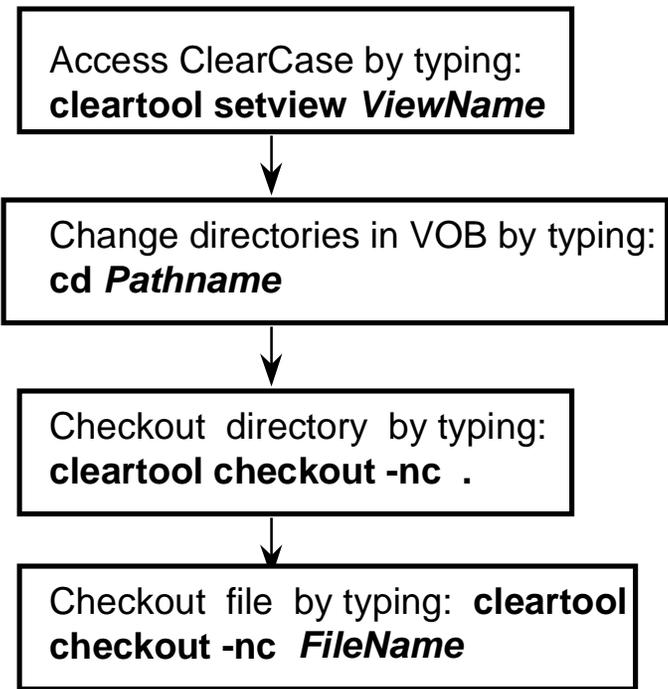
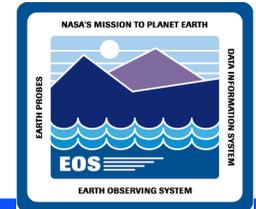
Key Assumptions

- A VOB and subdirectory has been created to hold the file.
- A view has been created.

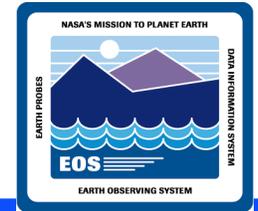
ClearCase Commands

- `cleartool setview ViewName` - Launches ClearCase and displays the user's view.
- `cd PathName` - Changes directory to a subdirectory in the VOB.
- `cleartool checkout -nc .` - Check out a directory from ClearCase
- `cleartool checkout -nc FileName` - Check out a file from ClearCase.
- `cleartool uncheckout` - Cancels a checkout.

Checking Out an Element from ClearCase



Entering a Modified Element into ClearCase



Key Assumptions

- A VOB and subdirectory has been created to hold the file.
- A view has been created.
- A file has been checked out from ClearCase and modified.

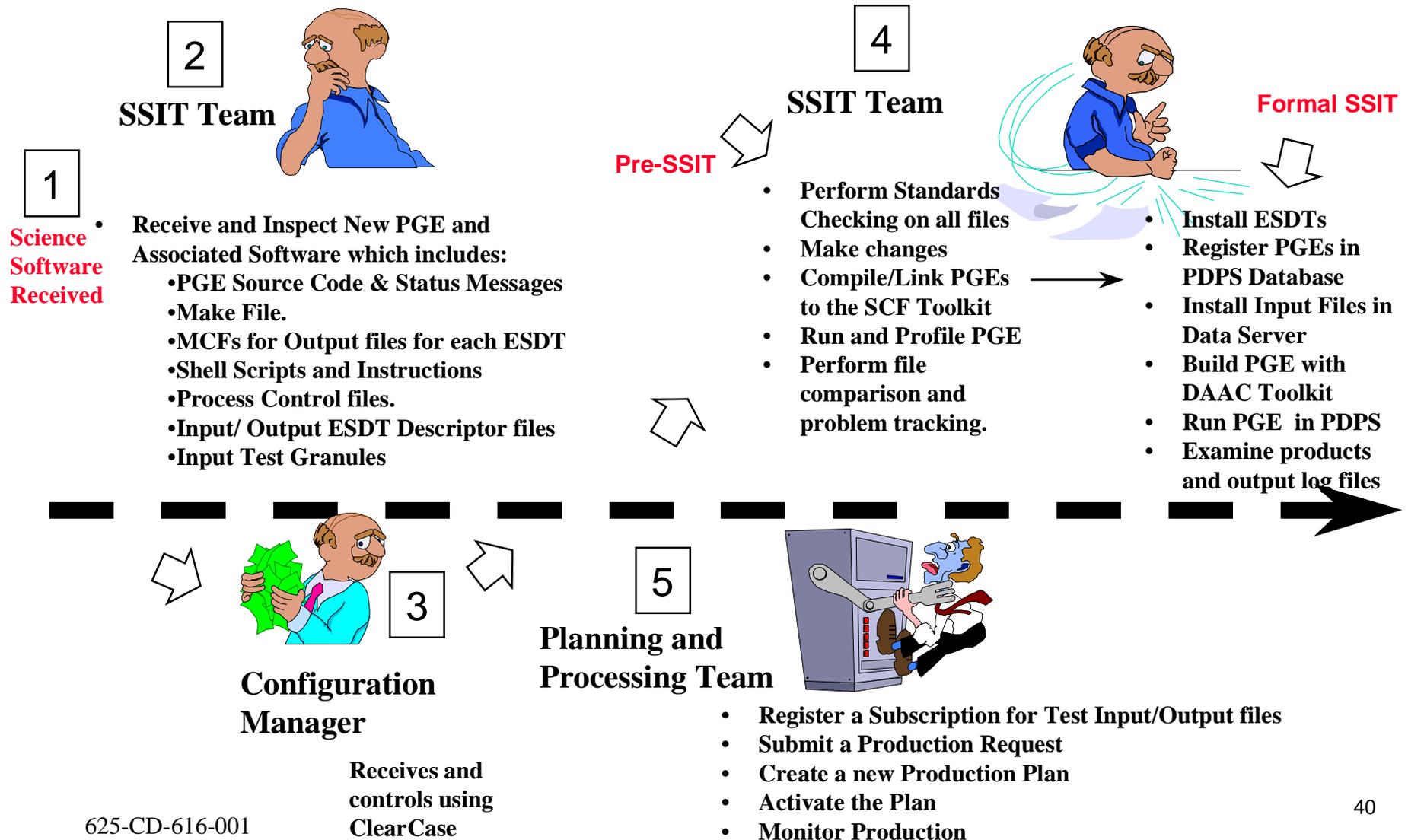
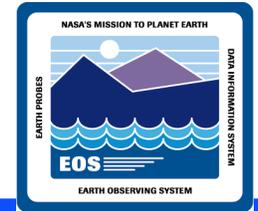
ClearCase Commands

- `cleartool setview ViewName` - Launches ClearCase and displays the user's view.
- `cd PathName` - Changes directory to a subdirectory in the VOB.
- `cleartool checkin -nc FileName` - Checks a modified file into ClearCase.

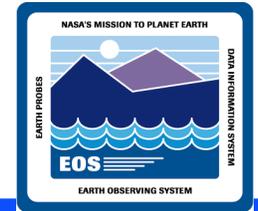
Note

- DAAC policy may require a comment on entry of modified element into ClearCase.

SSI&T Process Overview - The Big Picture

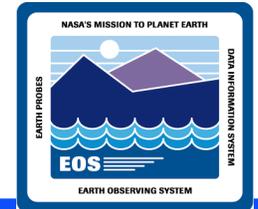


Science Office ECS Project Instructions



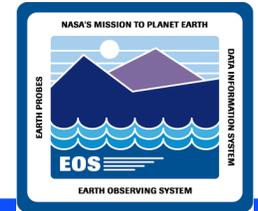
| | | |
|-----------------|--|----------------|
| SO-1-002 | Earth Science Data Type Generation Procedures PDF | 5/11/98 |
| SO-1-003 | Science Software Integration and Test (SSI&T) PDF | 9/14/98 |
| SO-1-004 | Science Office Science Support Internal Processes PDF, | |
| | Science Office Science Support Internal Processes (161-IT-003-001)PDF | 6/30/98 |
| SO-1-005 | Product Specific Attribute (PSA) Analysis PDF | 7/23/98 |

Science Office ECS Project Instructions Continued



| | |
|-----------------|---|
| 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 Validates Update Procedures PDF |
| 7/24/98 | |
| SO-1-011 | Metadata-Process Established with MODIS PDF |
| 8/20/98 | |

SSI&T PROCESS FLOW DIAGRAM 1

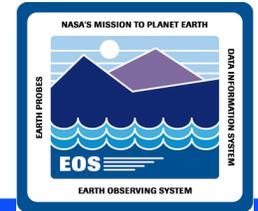


They can be accessed from the Web using the following URL:

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

Select: SO-1-003 and 004

Acquiring the Delivered Algorithm Package by FTP



Log into **sgi machine**

Type **cd *PathName***

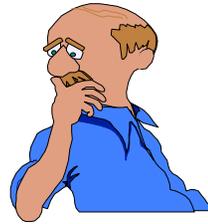
Type ***ftp machine***
IPaddress

Type ***user name***
and ***password***

Type **cd *PathName***
where DAP is located

Type **binary** and
get *DAPFilename*

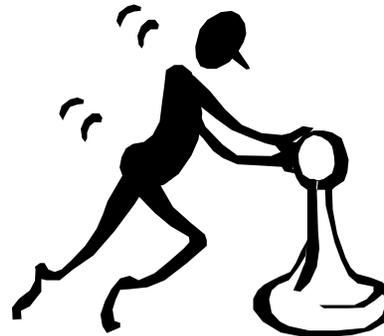
625-CD-616-001



SSIT Team

Note: Tapes are also used to transfer files. Usually performed by the SA.

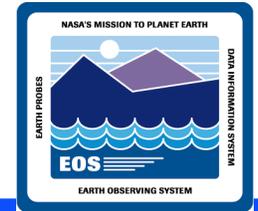
sgi machine
designated for SSI & T



FTP Transfers files to the desired directory within the **sgi machine**



Unpacking the Delivered Algorithm Package



Log into **sgi**
machine

Type **cd**
UnpackPathName

Type **tar xvz**
PackedDAP



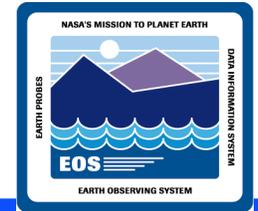
SSIT Team

The tar file is unpacked
in the desired directory
within the **sgi machine**.

DESTINATION: INGEST

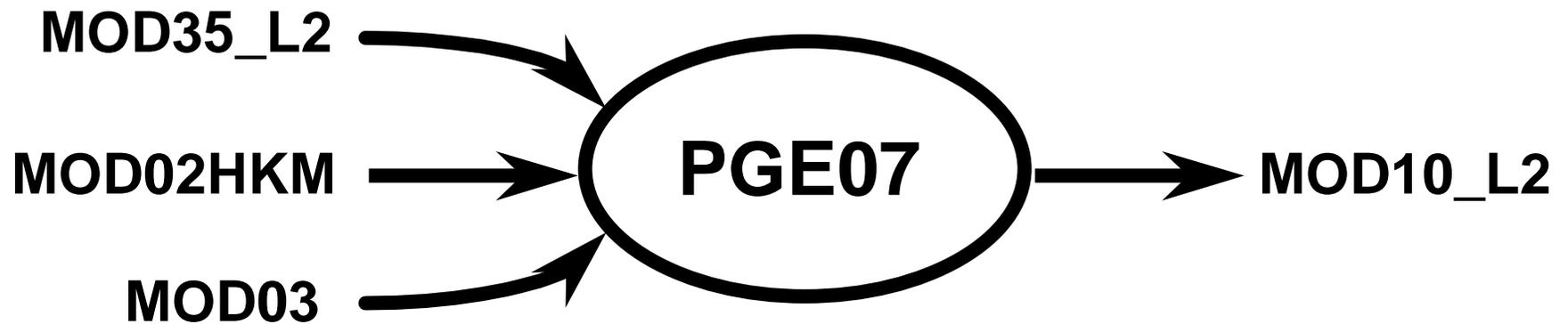
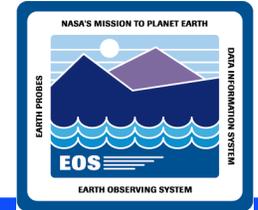


Delivered Algorithm Package



- **Typical Contents of DAP:**
 - **Source code**
 - **Message files.**
 - **Make or build files.**
 - **Shell Scripts**
 - **Process Control File (PCF).**
 - **Metadata Configuration File information to build an (MCF).**
 - **Instructions for building and running PGE.**
 - **New metadata and ESDT Descriptor files.**
 - **Test data for input and comparisons.**

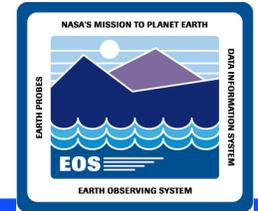
Training Example DAP MODIS PGE07



All files: 5 min data

Training Example DAP

(1 / 3)



Filename

Description

SDSRV:

DsESDTMoMOD03.001.desc

Input ESDT descriptor file

DsESDTMoMOD02HKM.001.desc

Input ESDT descriptor file

DsESDTMoMOD35_L2.001.desc

Input ESDT descriptor file

DSESDTMoMOD10_L2.001.desc

Output ESDT descriptor file

libDsESDTMoMOD03.001.Sh.so

Shared library for input ESDT

libDsESDTMoMOD02HKM.001.Sh.so

Shared library for input ESDT

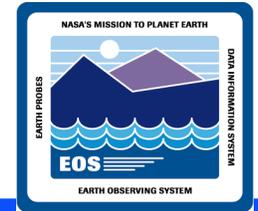
libDsESDTMoMOD35_L2.001.Sh.so

Shared library for input ESDT

libDsESDTMoMOD10_L2.001.Sh.so

Shared library for output ESDT

Training Example DAP (2 / 3)



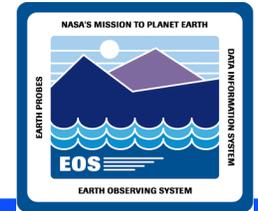
Filename

Description

PDPS:

| | |
|--------------------------------|------------------------------------|
| PGE07.tar | PGE executable |
| PGE07.tar.met | Target MCF for PGE executable |
| PGE_PGE07#1.0#01.odl | ODL for PGE07 |
| ESDT_ MOD03#2.0.odl | ODL file for binary input granule |
| ESDT_ MOD02HKM#2.0.odl | ODL file for binary input granule |
| ESDT_ MOD35#2.0.odl | ODL file for binary input granule |
| ESDT_ MOD10_L2#2.0.odl | ODL file for binary output granule |
| MOD02HKM.A1996218.1555.002.hdf | Binary input data granule |
| MOD03.A1996218.1555.002.hdf | Binary input data granule |
| MOD35_L2.A1996218.1555.002.hdf | Binary input data granule |

Training Example DAP (3 / 3)



Filename

Description

To be generated at run time:

MOD02HKM.A1996218.1555.002.hdf.met

Target MCF for binary input granule

MOD03.A1996218.1555.002.hdf.met

Target MCF for binary input granule

MOD35_L2.A1996218.1555.002.hdf.met

Target MCF for binary input granule

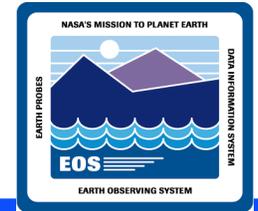
MOD10_L2.A1996218.1555.002.mcf

MCF for output product

MOD10_L2.A1996218.1555.002.mcf.met

Target MCF for status

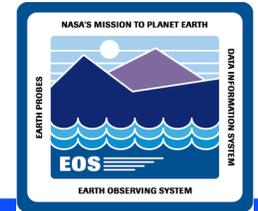
SSIT Manager



Provides a common interface to the SSI&T software tools and manages their operation

- **Setup SSI&T Manager and checklist.**
- **Open xterm session.**
- **Code Analysis.**
- **Office Automation Tools.**
- **Standards Compliance.**
- **Product Examination using EOSView and IDL.**
- **File Comparison in HDF, binary or ASCII format.**
- **Edit Text file.**
- **Initialize and Update PDPS database.**
- **Data Server Access.**
- **Additional information on SSI&T tools used by the DAAC operations specialist are found in Version 2.0 Operations Tools Manual 609-CD-003-003, section 4.5..**

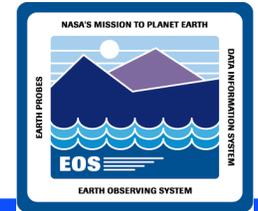
Setup of SSIT Manager



Configuration of Environment

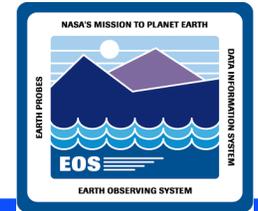
- **SSIT Manager runs only on Sun platforms in the subsystem DPS**
- **User makes a local copy of a Process Control File (PCF) for SSIT Manager and repeat same for a sample checklist.**
- **To setup the environment for the SSIT Manger, execute the procedures steps tht follow: Note that in the following, EcsCustomSW is normally in the directory /usr/ecs/TS1/CUSTOM on the AIT Sun: (This procedure was tested by telnet t1ais01, ID:, pw: At a UNIX prompt on an AIT Sun, (type**
 - setenv ECS_HOME /usr/ecs**
 - **cp /usr/ecs/{mode}/CUSTOM/data/DPS/EcDpAt.pcf**
 - **\$HOME/mySSITpcf, press Return.**
 - **setenv PGS_PC_INFO_FILE \$HOME/mySSITpcf, press Return.**
 - **check env for proper home path.**

Setup of SSIT Manager Continued



- type `cd /usr/ecs/mode/CUSTOM/utilities`, press Return.
- setenv `<mode>`
- Type in: `EcDpAtMgrStart <mode> &`
- This invokes the SSIT Manager GUI which should be displayed.

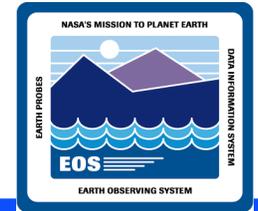
Setup of Checklist for SSIT Manager



Steps to Setup the SSIT Manager Checklist for Use in SSI&T

- From the SSIT Manager, click on the Tools menu, then choose Product Examination, then EOSView. The EOSView GUI will be displayed.
- Additional procedures are listed in the Training Manual vol 16.

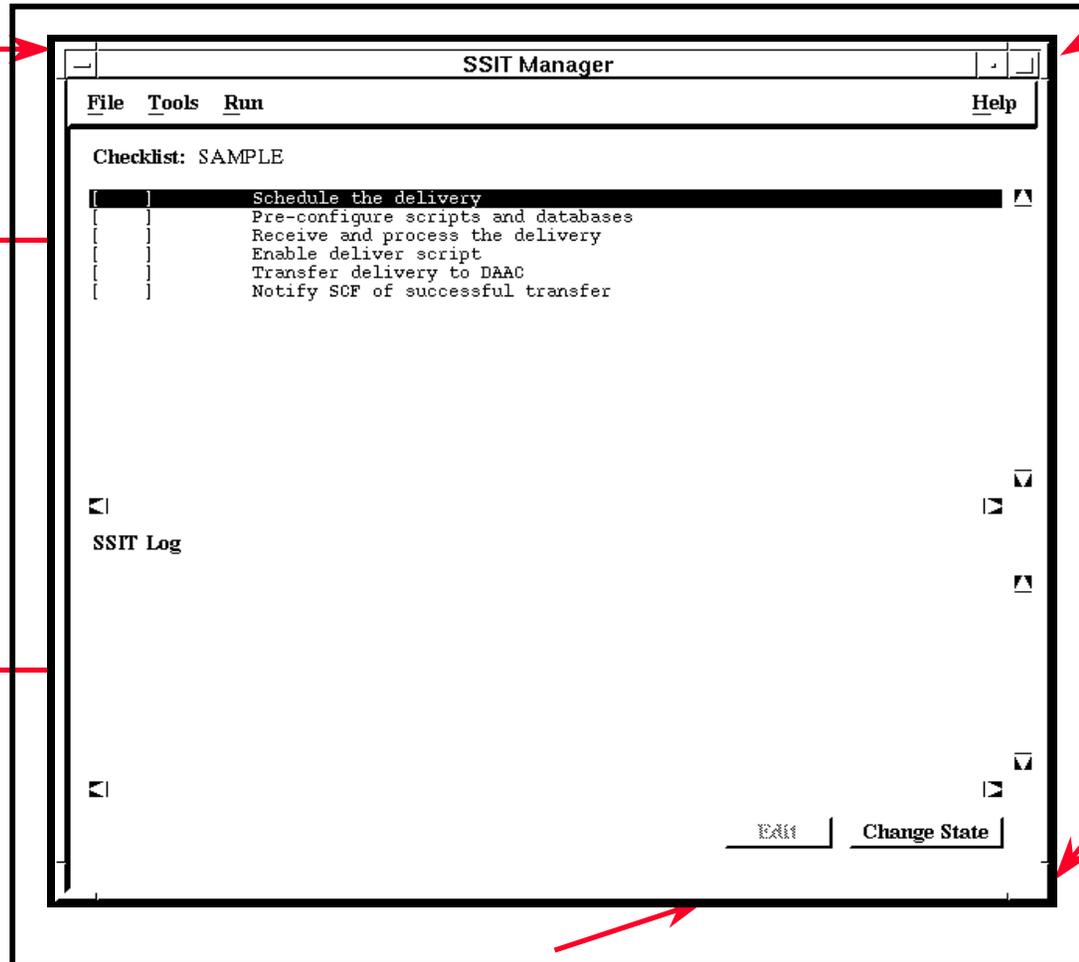
SSIT Manager GUI



Menu Bar:
Allows access to SSI&T Tools

Checklist Pane:
List set of steps to be completed

Log Pane:
Log of activities accomplished

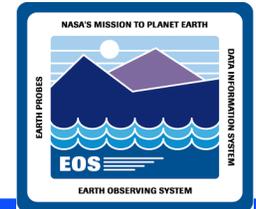


Help: Provides access to help features

Change State: Button allows Checklist state to Toggle

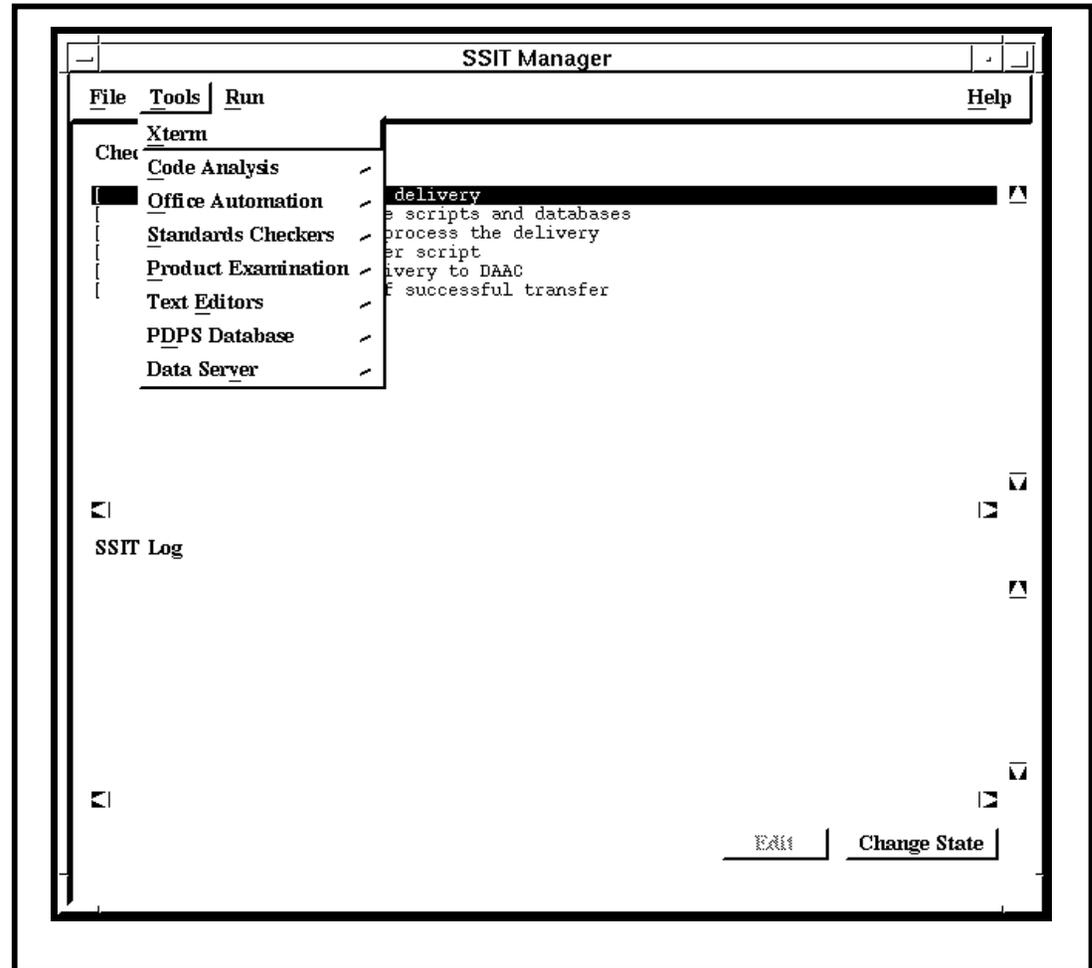
Edit: Button allows Checklist to be edited

SSIT Manager Tools

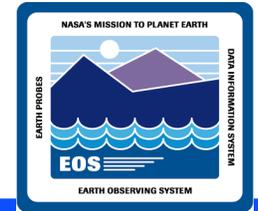


Tools:

1. **Xterm:** Starts an Xterm window session
2. **Code Analysis:** Performs static code analysis
3. **Office Automation:** MS Windows, MS Office, MS Excel
4. **Standards Checkers:** Prohibited Function & Process Control File Checkers, ForCheck, Prolog Extractor
5. **Product Examination:** File Comparison Tools and EOSView
6. **Text Editors:** Emacs or Xedit Tools
7. **PDPS Database:** PCF ODL Template Tool, Science Metadata & Opnl Metadata Update
8. **Data Server:** Register Subscription, Insert Static, Insert Test Dynamic, Insert EXE TAR



Standards Checking

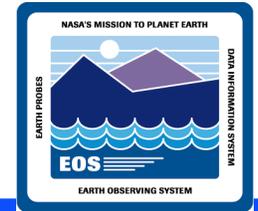


Purpose: Verify that the source files of a PGE are compliant with the ESDIS Data Production SCF Standards and Guidelines.

Key Terms:

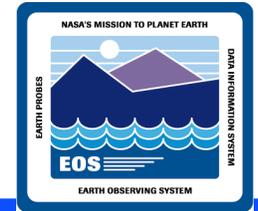
- **SDP Toolkit**
 - **provides an interface to the ECS system.**
 - **allows science software to be portable to different platforms.**
 - **reduces redundant coding at the SCF.**
 - **provides value added functionality for science software development.**

Standards Checking (cont.)



- **Mandatory SDP Toolkit Functions**
 - **Error and Status Message Facility (SMF).**
 - **Process control Tools.**
 - **Generic Input/Output Tools.**
 - **Memory Allocation Tools.**
- **Optional SDP Toolkit Functions**
 - **Ancillary Data Access.**
 - **Celestial Body Position Coordinate System Conversion.**
 - **Constant and Unit Conversion.**
 - **IMSL.**

Standards Checking (cont.)



Steps for Standards Compliance

FORTRAN 77 - On the AIT Sun.

Source FORCHECK setup file.

Create FORCHECK run script.

Invoke FORCHECK run script.

Examine the list file.

Fortran 90 and C - On the SDPS SGI.

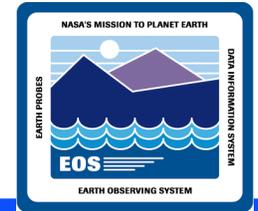
Set environment to appropriate SDP Toolkit.

Compile the PGE using compiler flags.

Examine the list file.

Ada - Compile using COTS Verdix Ada Development System or GNU C Compiler, gcc.

Prohibited Function Checker



•Used to check source files for the occurrence of functions that are prohibited in the ECS DAAC production environment.

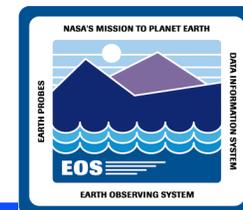
Key Procedure Commands

- **SSIT Manager**
 - **Tools → Standards Checkers → Prohibited Function Checker**

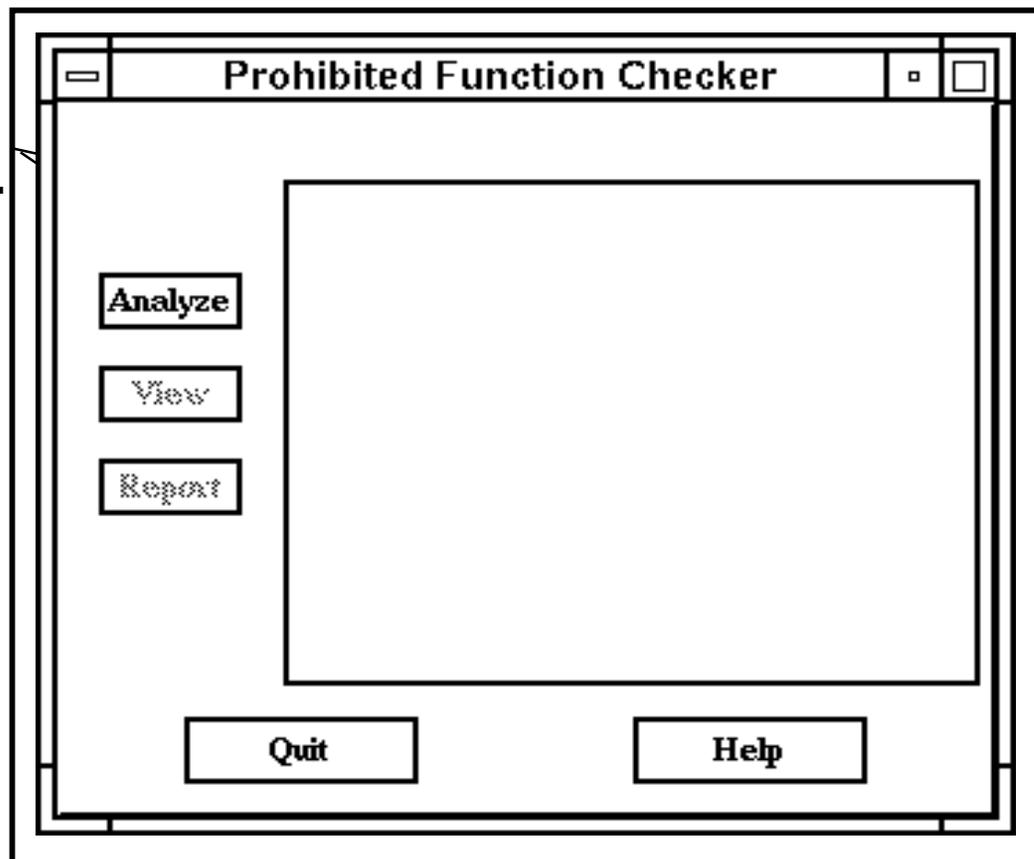
Run the Analyze from GUI.

- **Highlight files to be analyzed.**
- **Run checker.**
- **Generate report.**
- **Save and examine report.**

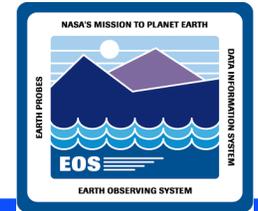
Prohibited Function Checker GUI



Highlight files to be analyzed.
Run checker.
Generate report.



Checking Process Control Files



Key Procedure Commands.

- **SSIT Manager**
 - **Tools → Standards Checker → Process Control File Checker**

Run the PCF Checker GUI.

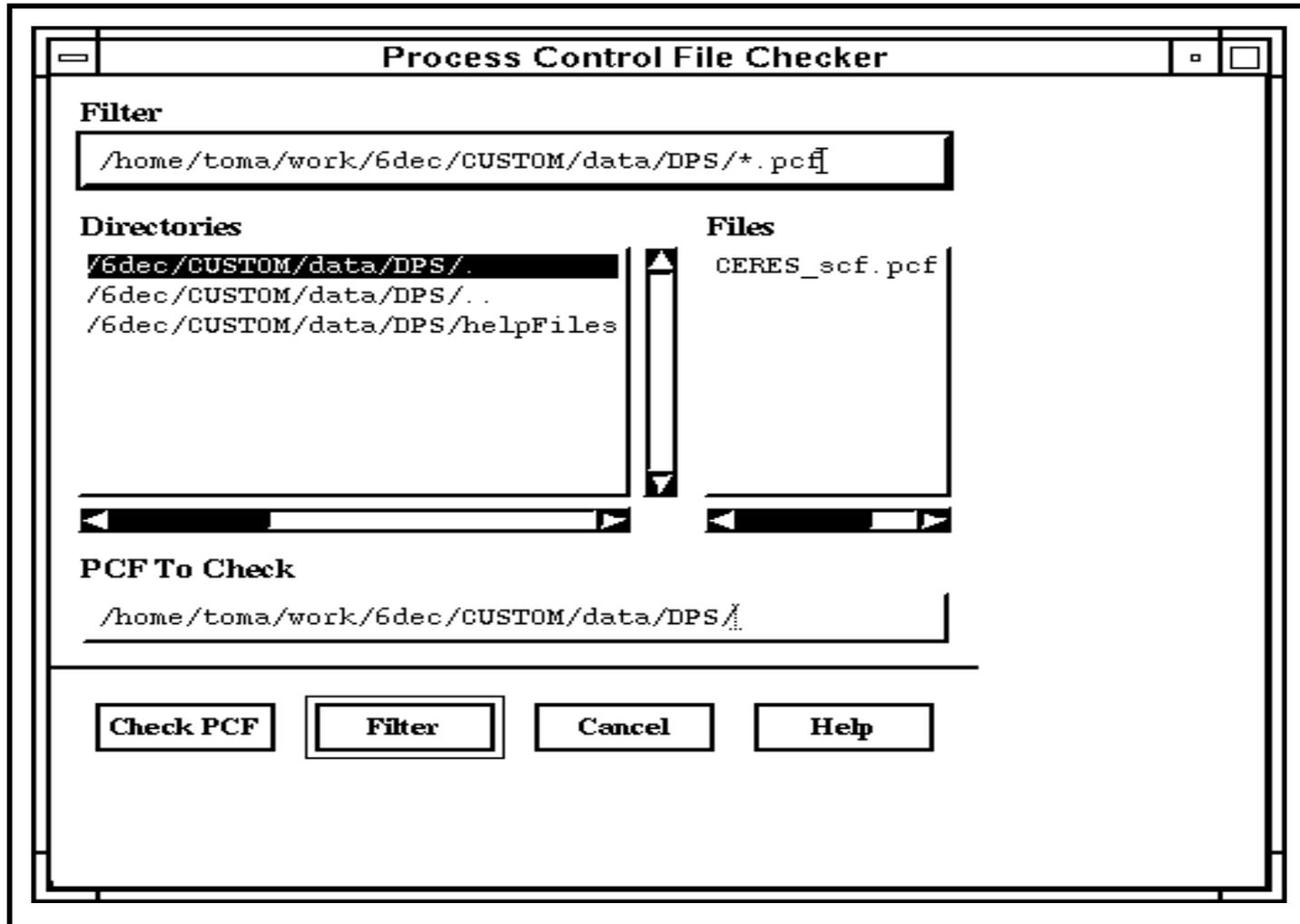
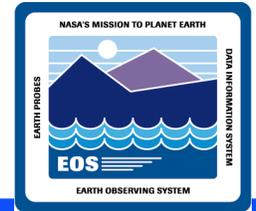
- **Select the directory**

The PCFs must be checked to verify that they are syntactically correct and contain all the information for the PGEs to run within the ECS DAAC production environment.

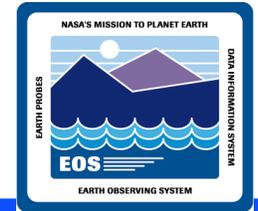
Select one PCF and select the Check PCF button.

- **Save or print the results file and examine results.**

Checking Process Control Files



Extracting Prologs



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

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

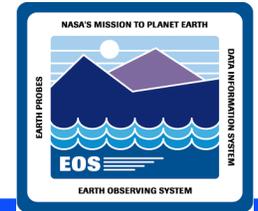
Key Procedure Commands:

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

Run the Prolog Extractor GUI.

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

Compiling and Linking Science Software

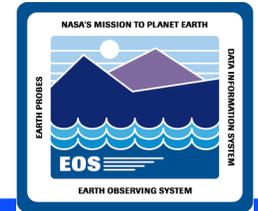


Science software developed at SCF's using the SDP Toolkit provided by ECS needs to be compiled and linked first with SCF Toolkit version to compare results at each facility. Then the science software needs to be compiled and linked with the DAAC Toolkit.

Preparation for compile and link:

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

Updating a PCF



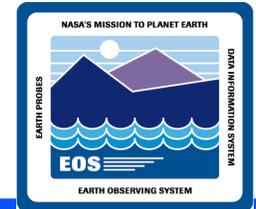
PCF sections:

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

Update appropriate path names where necessary:

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

Updating a PCF



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

Enter the vi editor by typing:
vi *PCFFFileName*

Edit the PCF.

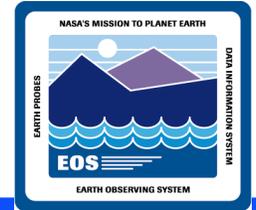
After editing, re-run the PCF
Checker by selecting
**Tools→Standards Checker→PCF
Checker** and from the SSIT menu.

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

Checkin the directory by typing:
cleartool checkin -nc .

A screenshot of a terminal window titled "Terminal". The window has a menu bar with "Window", "Edit", "Options", and "Help". The terminal prompt is "p1nt2sun: rfuller: /home/rfuller{133}>". Below the prompt, the text "> command line input" is displayed, followed by three lines of ">" characters, indicating that the user has entered commands but they are not visible in the screenshot.

Compiling the SMF



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

Provides:

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

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

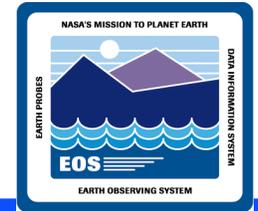
These files will be used by science software during runtime.

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

Process Steps:

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

Compiling a PGE and Linking with SCF and DAAC SDP Toolkits



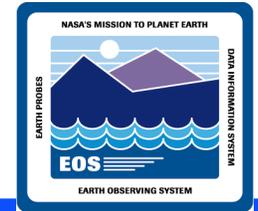
Compiling and Linking of Science Software will vary according to the instructions from the Instrument Software Development Teams.

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

Process Steps:

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

Running a PGE in a Simulated SCF Environment

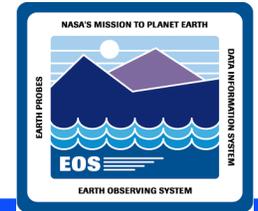


Running a PGE at the DAAC in a simulated SCF environment should produce identical results as those at the SCF.

Process Steps:

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

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

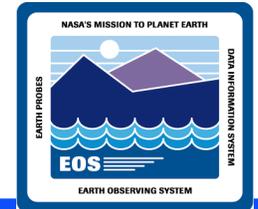


To capture PGE runtime statistics for the PDPS Database, perform profiling using the DpPrRusage Program (Rusage).

Statistics needed:

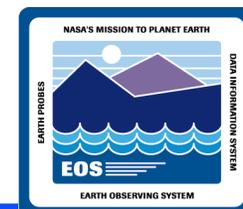
- wall clock time
- user time
- system time
- amount of memory used
- number of page faults
- number of input and output blocks
- number of swaps

Available SDP Toolkits used by the SGI science processors



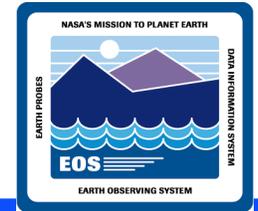
| SDP Version | Language Type | Library Object Type | \$PGSBIN |
|-------------|-----------------|---------------------|---|
| SCF | C++ or C | Old 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_scf_cpp |
| SCF | FORTRAN 77 or C | Old 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_scf_f77 |
| SCF | Fortran 90 or C | Old 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_scf_f90 |
| SCF | Thread | Old 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_scf_r |
| SCF | C++ or C | New 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_scf_cpp |
| SCF | FORTRAN 77 or C | New 32-bit Mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_scf_f77 |
| SCF | Fortran 90 or C | New 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_scf_f90 |
| SCF | Thread | New 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32__scf_r |
| SCF | C++ or C | 64-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_scf_cpp |
| SCF | FORTRAN 77 or C | 64-bit Mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_scf_f77 |
| SCF | Fortran 90 or C | 64-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_scf_f90 |

Available SDP Toolkits used by the SGI science processors continued



| | | | |
|------|-----------------|-----------------|--|
| SCF | Thread | 64-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_scf_r |
| DAAC | C++ or C | Old 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_daac_cpp |
| DAAC | FORTRAN 77 or C | Old 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_daac_f77 |
| DAAC | Fortran 90 or C | Old 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_daac_f90 |
| DAAC | Thread | Old 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_daac_r |
| DAAC | C++ or C | New 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_daac_cpp |
| DAAC | Fortran 77 or C | New 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_daac_f77 |
| DAAC | Fortran 90 or C | New 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_daac_f90 |
| DAAC | Thread | New 32-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_daac_r |
| DAAC | C++ or C | 64-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_daac_cpp |
| DAAC | Fortran 77 or C | 64-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_daac_f77 |
| DAAC | Fortran 90 or C | 64-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_daac_f90 |
| DAAC | Thread | 64-bit mode | \$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_daac_r |

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



Sample of an Rusage File produced:

lasher{emcleod}6: more Profile.out

```
source .cshrc
```

```
# cd TEST/MOD*
```

```
# ls
```

```
#
```

```
  /usr/ecs/OPS/CUSTOM/bin/DPS/EcDpPrRusa
```

```
ge MOD_PR10.exe > Profile.out
```

lasher{emcleod}9: more profile.out

```
# Resource Usage Information
```

```
COMMAND=MOD_PR10.exe
```

```
EXIT_STATUS=0
```

```
ELAPSED_TIME=233.583145
```

```
USER_TIME=10.046158
```

```
SYSTEM_TIME=7.555547
```

```
MAXIMUM_RESIDENT_SET_SIZE=4080
```

```
AVERAGE_SHARED_TEXT_SIZE=0
```

```
AVERAGE_UNSHARED_DATA_SIZE=0
```

```
AVERAGE_UNSHARED_STACK_SIZE=0
```

? PAGE_RECLAIMS=151

? PAGE_FAULTS=0

? SWAPS=0

? BLOCK_INPUT_OPERATIONS=2

? BLOCK_OUTPUT_OPERATIONS=27
10

? MESSAGES_SENT=0

? MESSAGES_RECEIVED=0

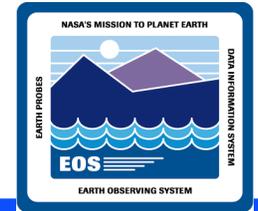
? SIGNALS_RECEIVED=0

? VOLUNTARY_CONTEXT_SWITCHES=1095

? INVOLUNTARY_CONTEXT_SWITCHES=2

? lasher{emcleod}10:

Examining PGE Produced Log Files



PGEs produce three log files during runtime:

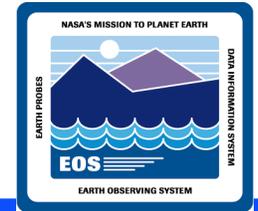
- **Status Log - captures all error and status information.**
- **User Log - captures a subset of more informational messages.**
- **Report Log - captures arbitrary message strings.**

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

Process Steps:

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

File Comparison and Data Visualization



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

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

Searches are performed for any differences beyond specified tolerances.

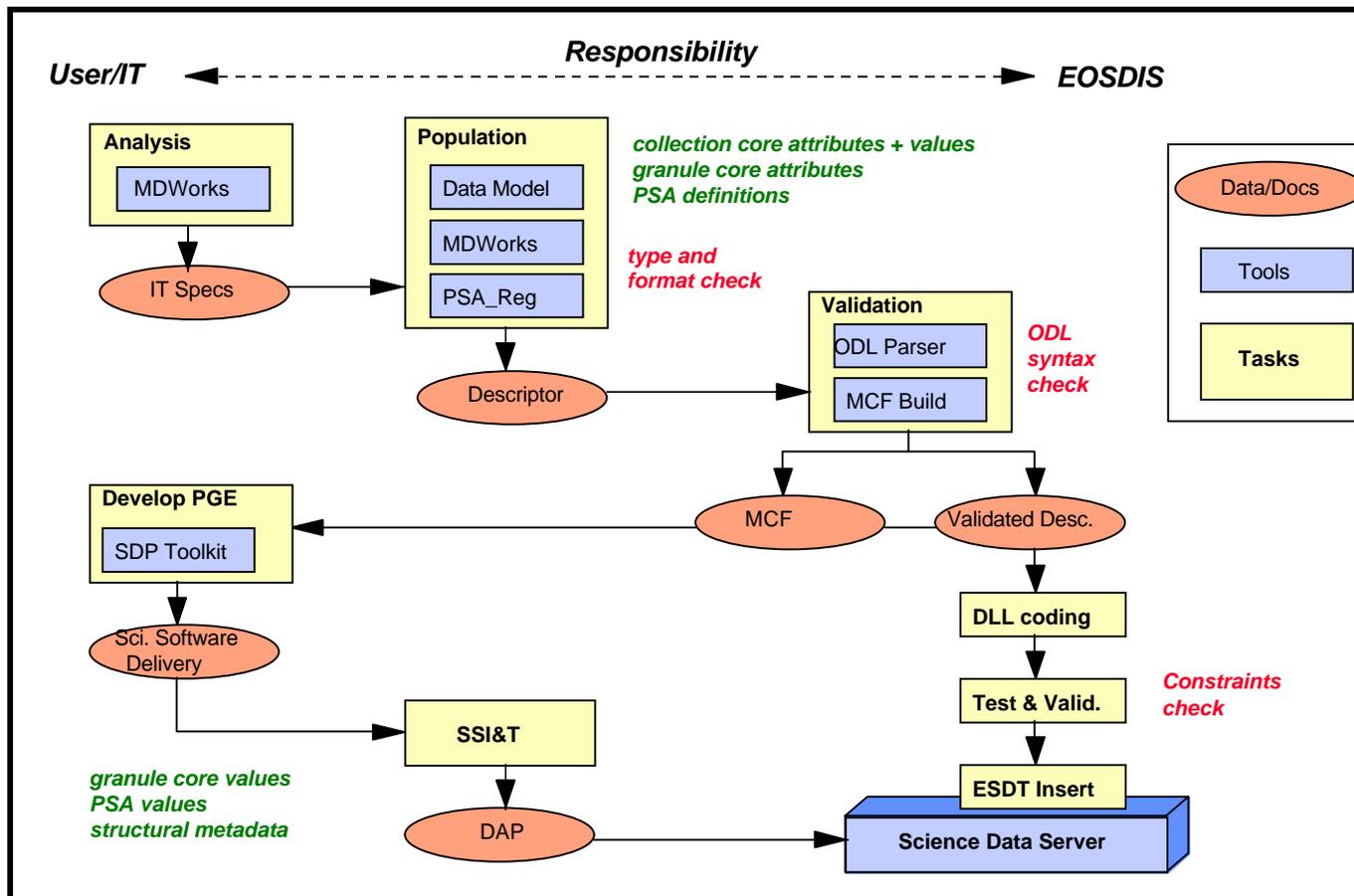
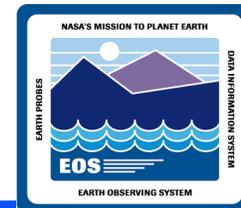
Data product files can be compared by a variety of tools accessible by the SSIT Manager GUI.

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

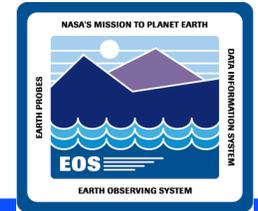
Data visualization tools are accessible by the SSIT Manager GUI.

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

The Metadata Workflow

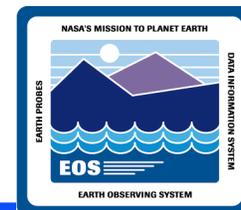


Earth Science Data Types- ESDT's



- **Representation of different types of data products from the scientists perspective.**
- **Define to the Version 2 Data Model:**
 - **Collection level metadata attributes and values.**
 - **Granule level metadata attributes.**
 - **Data services appropriate to the ESDT collection and data granules within the collection.**

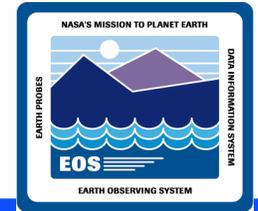
ESDT Components



Descriptor File:

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

Preparation of Earth Science Data Types (ESDTs)

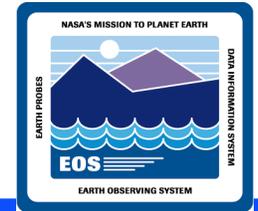


Building and installing ESDTs has been done before ECS is applied.

ECS Requirements:

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

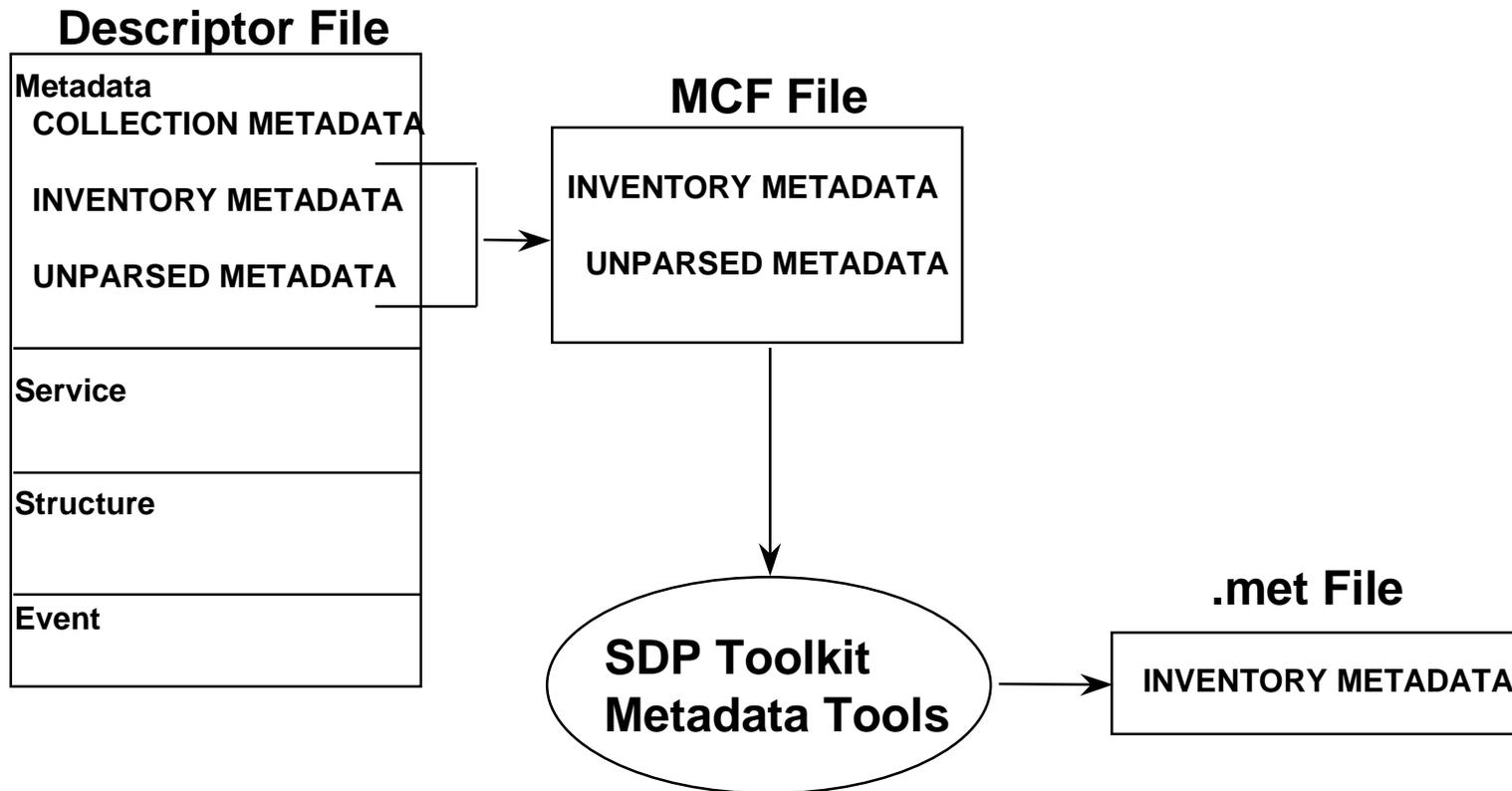
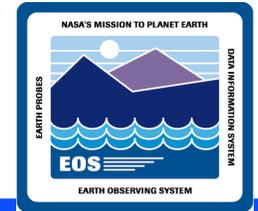
MCF Generation from SDSRV



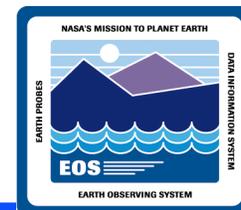
The Metadata Configuration File (MCF) is produced by the following processes:

- The ESDT's and DLL's are installed into the SDSRV with error checking taking place in the descriptor before installation.
- The GETMCF tool is executed to pull the MCF from the SDSRV. The actual MCF is generated and then copied from the Inventory and Archive section of the Collection Descriptor.
-

Descriptor, MCF & .met Files

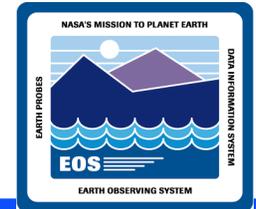


What are Validations?



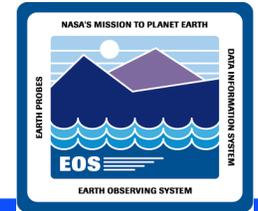
- **Validations are type of integrity constraint to ensure that metadata values comply with the data model and database schema requirements**
- **Validations are used to ensure the data products quality and consistency of search queries**
- **Currently implemented for**
 - **Data Type and Length checking; e.g. *STRING, FLOAT, etc.***
 - **Match Rules; e.g. *DayNightFlag = (Day, Night, Both)* ex: *DAY* will not match, resulting in a failed insert. (match rule is exacting)**
 - **Range Checking ; e.g.**
 - Longitude Minimum = -180.00 to**
 - Longitude Maximum = +180.00**
 - **Expressions; e.g. *MinimumAltitude >= 0.0***

Attribute Valid Processing



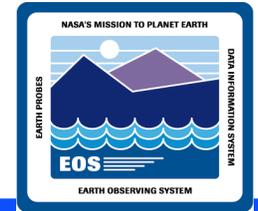
- **Attribute Valid**s have dependency with corresponding **Descriptors, PGEs; Versions must be “in-sync”**
 - **Attribute Valid**s values may have a dependency upon **SDSRV** version and potentially other subsystems(Client & Ingest for example)
 - **Changes to Valids may impact code; Versions must be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

Attribute Valid Processing (Cont'd)



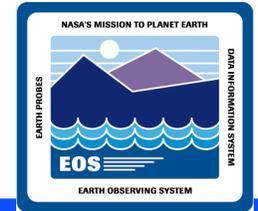
- **Currently, incoming metadata checked against one of the 4 basic constraints checks (Match, Range, Expression, or NONE) as specified**
- **“NONE” means no value checking is performed**
- **Messages are logged for attributes containing invalid data. Action on ESDT/granule with invalid metadata depends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

PSA Process



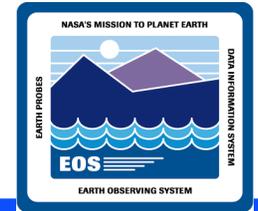
- **Product Specific Attributes (PSA) information is obtained from Instrument Teams through a template (PSA Template) that has been provided to them**
- **Data Engineering performs analysis to verify that the PSAs are unique and conform to the Data Model**
- **Approved PSAs are submitted to the ECS CCB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

ESDT CM Process



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

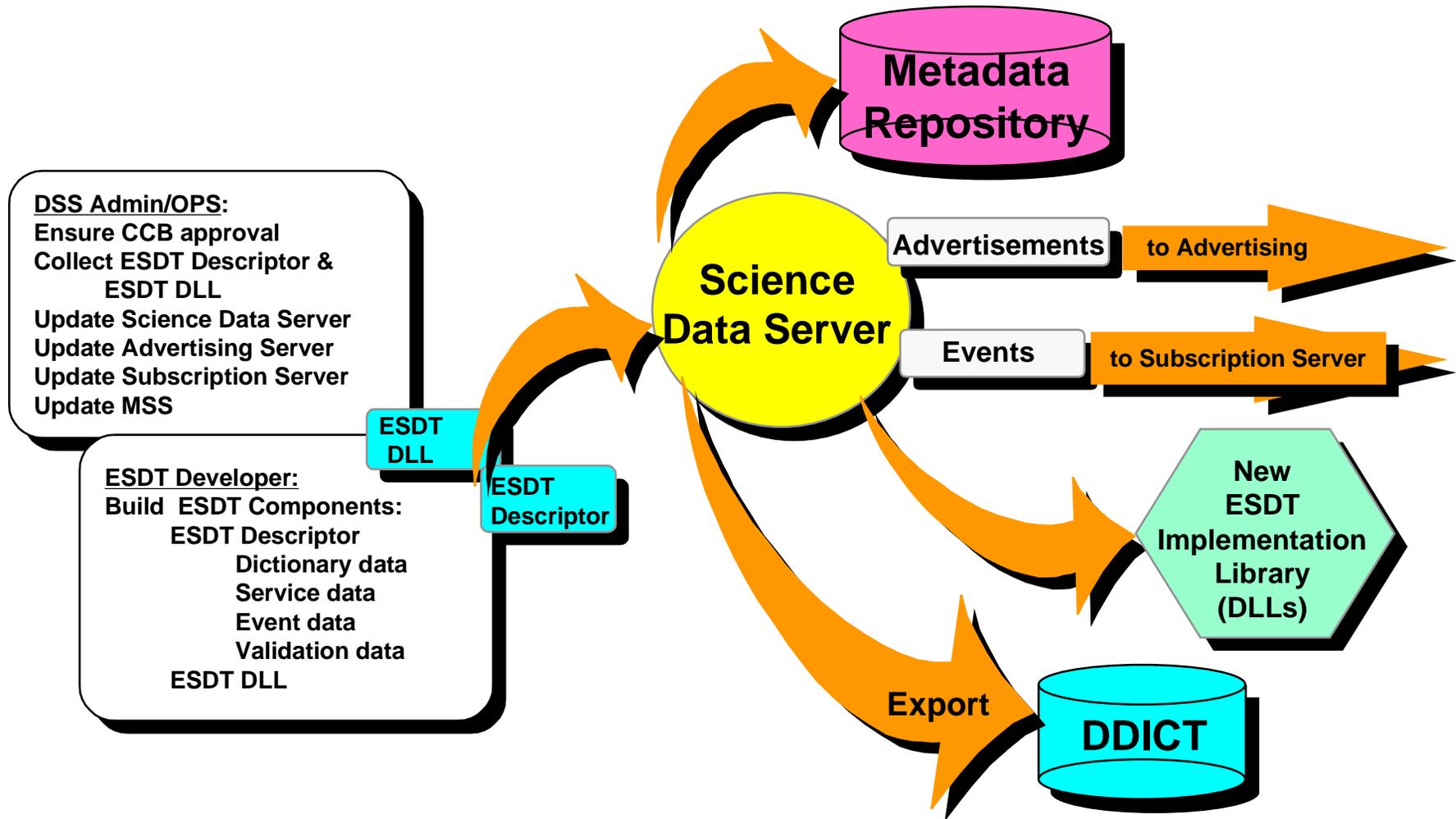
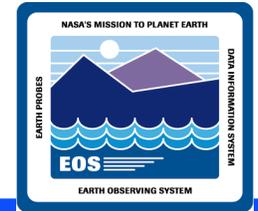
Overview: ESDT Development and Installation Process



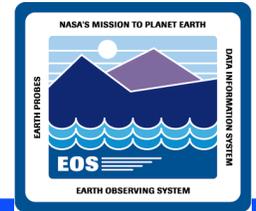
Complete ESDT development and installation process involves the following steps:

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

Adding a new ESDT: Operational Overview

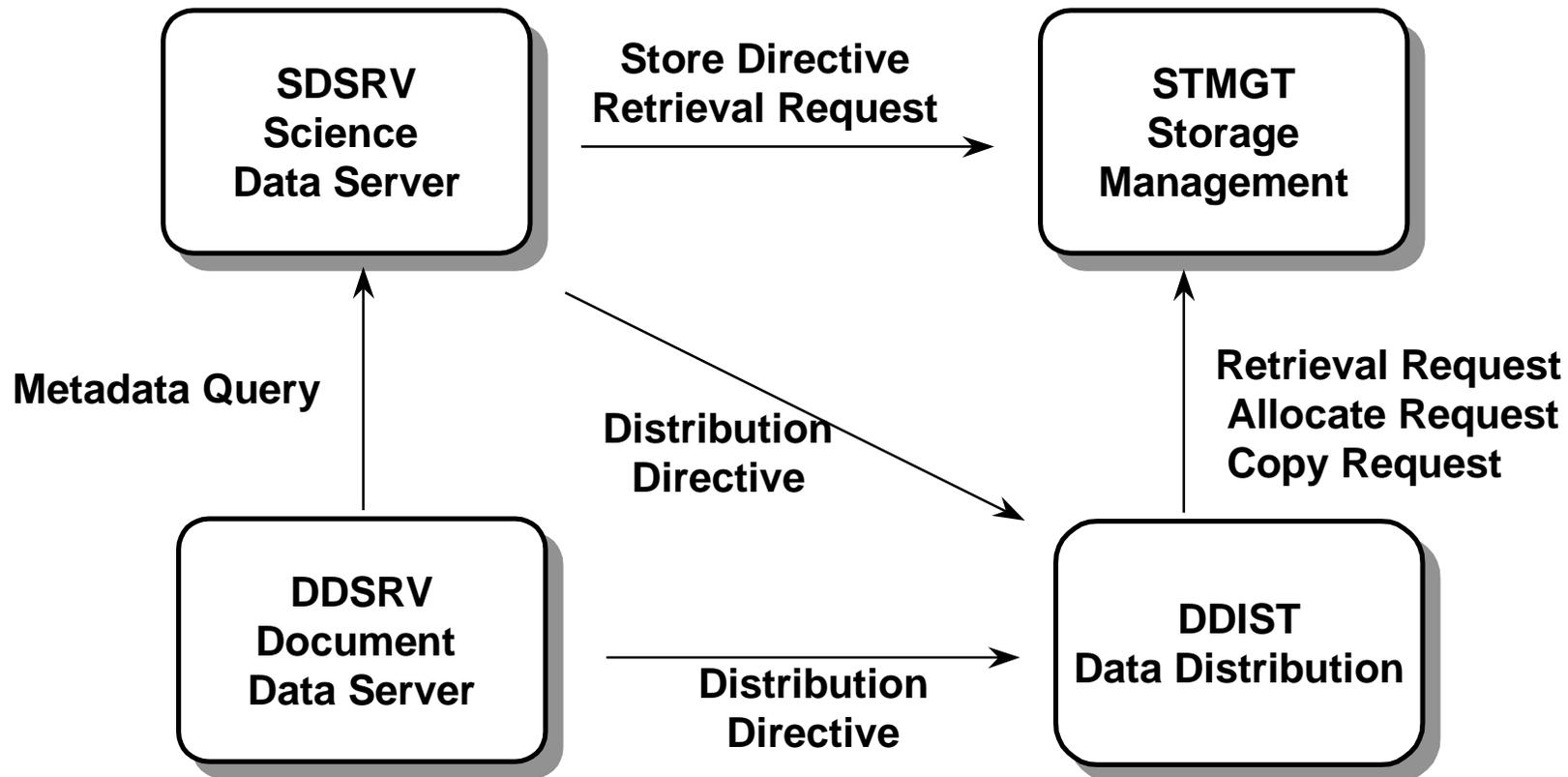
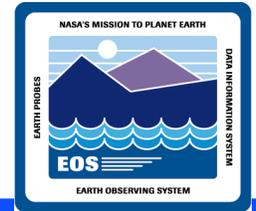


Required Servers

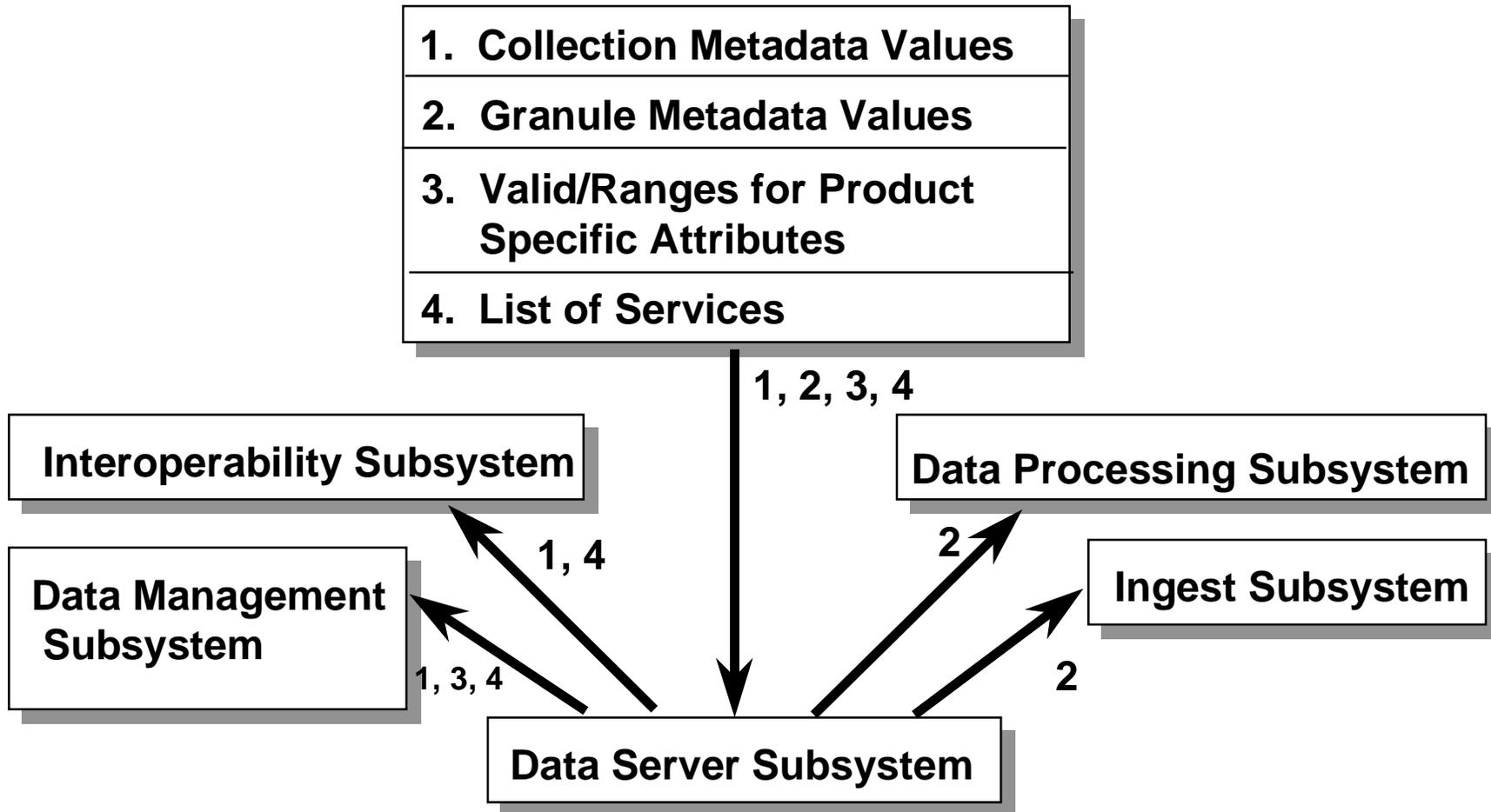
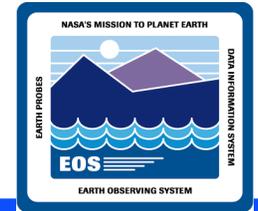


- **Following servers need to be started and running before installing ESDTs (with GDAAC machine names as examples):**
 - **Science Data Server (SDSRV) (g0acs03)**
 - **Storage Management Servers (STMGT) (g0icg01, g0drg01, g0dps02)**
 - **Data Distribution Servers (DDIST)(g0dps02)**
 - **Subscription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

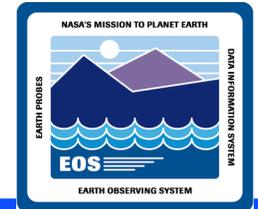
Data Server Subsystem: Software Components



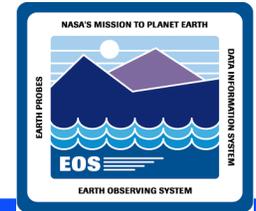
Metadata Flow in ECS



ECS ASSISTANT



Bringing Up ECS Assistant



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

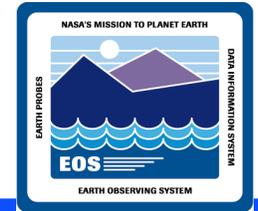
1 At the UNIX Console or Terminal type **setenv DISPLAY:0.0**,
To verify the setting, type **echo \$DISPLAY**, press **Enter**.

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

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

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

Bringing Up ECS Assistant continued



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

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

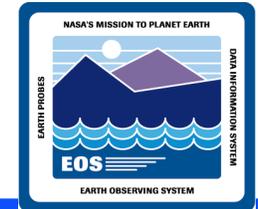
A ClearCase view is required only if the ECS Assistant needs to be able to “see” into a ClearCase VOB; a view is not necessary otherwise.

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

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

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

ESDT Manager GUI

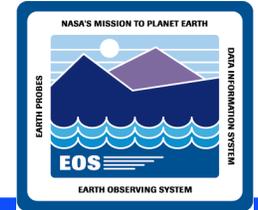


The screenshot displays the ESDT Manager GUI with the following components:

- File Latest Info** (top left)
- Look In** dropdown set to **Clearcase**, **Mode** dropdown set to **RCCCO**, and a **DB Viewer** button.
- Clearcase Directory** and **Mode Directory** input fields.
- DLL Path** input field.
- Buttons: **Configure**, **Verify**, **Add**, **Remove**.
- Navigation buttons: **As**, **Ce**, **Ea**, **Ed**, **Ls**, **Mi**, **Mo**, **Mp**, **Sa**, **Sys**.
- Descriptor Files <161>** list:
 - DsESDTCeCER00AA.desc
 - DsESDTCeCER00AA.X.desc
 - DsESDTCeCER00AF.desc
 - DsESDTCeCER00AFX.desc
 - DsESDTCeCERCALAA.desc
 - DsESDTCeCERCALAF.desc
 - DsESDTCeCERCLAA.X.desc
 - DsESDTCeCERCLAFX.desc
 - DsESDTCeCERDIAA.desc
 - DsESDTCeCERDIAA.X.desc
 - DsESDTCeCERDIAF.desc
 - DsESDTCeCERDIAFX.desc
 - DsESDTCeCBANALAA.desc
 - DsESDTCeCBBSDSAB.desc
 - DsESDTCeCBBDSAB.desc
 - DsESDTCeCBBDS_AB.desc
- Selected Files <1>** list:
 - CER00AA
- Buttons between panes: **All**, **==>**.
- Buttons at bottom of panes: **View File**, **Remove**, **Clear All**.
- Results** pane showing terminal output:

```
/usr/ecs/RCCCO/CUSTOM/utilities/EcDsSrAdesdt RCCCO /usr/ecs/RCCCO/CUSTOM/lib/ESS/ CER00AA
Re-run after setting the following environment variables:
DBPASSWD
```
- Current Directory:** /usr/ecs/RCCCO/CUSTOM/bin/DSS/.
- Directory Statistics:** drwxrwxrwx 5 adupree users 60416 Nov 19 16:27
- GUI Information** (bottom right)

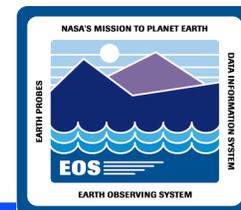
Installing an ESDT/DLL using Science Data Server Operator GUI



Key Assumptions

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

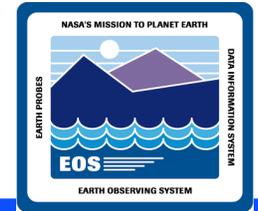
Installing an ESDT/DLL using Science Data Server Operator GUI



The screenshot shows the "Science Dataserver Operator GUI - [DEV04]" window. It has a menu bar with "File", "Selected", "Options", and "Help". Below the menu bar are two icons: "Data Types" (a printer) and "System Requests" (two computers). The main area is titled "Science Data Server - Data Types" and contains a "Data Type Information" table. Below the table is a "Find" search box. At the bottom are four buttons: "View", "Add...", "Update...", and "Refresh/Reconnect". The "Operator Messages" section at the very bottom is currently empty.

| ID | Name | Version | Description |
|-------------------------|----------|---------|-------------|
| SGI_DEV04: AIRI2IFC.001 | AIRI2IFC | 1 | |
| SGI_DEV04: AIRI2RED.001 | AIRI2RED | 1 | |
| SGI_DEV04: AIRI2TMC.001 | AIRI2TMC | 1 | |
| SGI_DEV04: AM1ANC.001 | AM1ANC | 1 | |
| SGI_DEV04: AM1ATTF.001 | AM1ATTF | 1 | |
| SGI_DEV04: AM1ATTH0.001 | AM1ATTH0 | 1 | |
| SGI_DEV04: AM1ATTHF.001 | AM1ATTHF | 1 | |
| SGI_DEV04: AM1ATTN0.001 | AM1ATTN0 | 1 | |
| SGI_DEV04: AM1ATTNF.001 | AM1ATTNF | 1 | |
| SGI_DEV04: AM1EPHF.001 | AM1EPHF | 1 | |
| SGI_DEV04: AM1EPHH0.001 | AM1EPHH0 | 1 | |
| SGI_DEV04: AM1EPHHF.001 | AM1EPHHF | 1 | |
| SGI_DEV04: AM1EPHNO.001 | AM1EPHNO | 1 | |
| SGI_DEV04: AM1EPHNF.001 | AM1EPHNF | 1 | |

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



Entering from Command Line

EcDsSdsrvGuiStart <MODE>. This will bring up a GUI where you can click on **ADD**. Another GUI will appear “**Add Data Type**”. Enter the following:

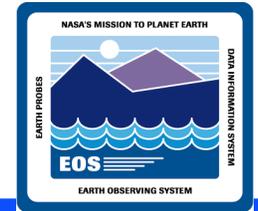
Descriptor Filename: Enter path to where ESDT is located, including the full ESDT descriptor.

click on: **OK**

If added successfully, another GUI will appear saying ...**Data Type Successfully Added**.

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

Viewing and Copying ESDT/DLL using ECS Assistant GUI

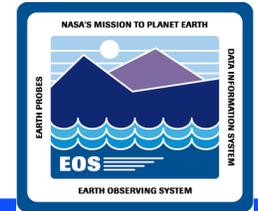


Key Assumptions

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

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

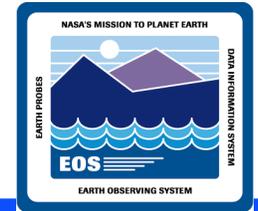
Validating Successful ESDT Installation



Criteria for success

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

Removing ESDT's Using the Command Line



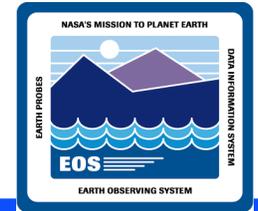
Removing ESDT's from 4 Data Bases, SDSRV, ADSRV, SUBSRV and DDICT using Command Line.

First delete ESDT's from the Science Data Subsystem:

Procedures:

- 1 telnet to (SDSRV) p0acs03[e.g.]**
- 2 login: , password:**
- 3 setenv DISPLAY.....:0.0**
- 4 cd /usr/ecs/TS1/CUSTOM/local**
- 5 cleansdt.csh <descriptor_file_name> (If an ESDT
has to be removed before replacing it)**

Removing ESDT's Using the Command Line continued



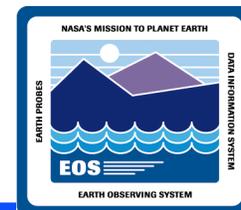
- The ESDT will be removed from all four servers, SDSRV, SUBSRV, ADSRV & DDICT. Check SDSRV Gui to see if removal was successful.

6 Now you have to log into each server and bring each of the four servers mentioned above, down by using the **KILL Script** and the **START Script** that have been specifically taylorred for each server at the path identified below in each subsystem:

cd/home/cmshared/bin/

- **SDSRV** you are already on, **p0acs03**, **SBSRV** is on **p0ins01**, **ADSRV & DDIC** are on **p0ins02**.

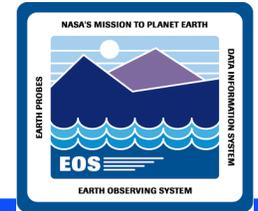
Staging ESDT's for Installation



- 1 Copy into a directory from which the Science Data Server will use to bring ESDT's into the Science Data Server Data Base
- 2 Once you have the descriptor file copied into `<your_directory>` in the **p0ins01 machine**, you have to login into **p0acs03** and copy the file into `/usr/ecs/TS1/CUSTOM/data/ESS` directory. To do so, log into **p0acs03**. From the directory `/usr/ecs/TS1/CUSTOM/data/ESS`, Type command,
cp /home/your_directory/ESDTs/filename .
(Where `/home/your_directory/ESDTs` is the directory into which you stored the descriptor file.)

Now you have the copy of the latest descriptor file in the **ESS** directory.

Installing ESDT's Using the Science Data Server GUI



- 1 Carryout the procedures in Removing ESDT's from Archive using Command Line Scripts section steps 1 through 5, then make the determination as to whether you need to remove existing ESDT's.

Now you are ready to start Installation procedures.

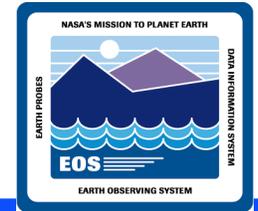
- 2 Once you know that all the 4 Science Data Servers are down, then log into **p0acs03** machine, go to directory **/usr/ecs/TS1/CUSTOM/utilities** and execute command

EcDsScienceDataServerStart TS1 StartTemperature cold

(This script will bring up the four Data Servers.)

- Then check the status of the servers by using a Server Monitor GUI WHAZZUP. Once all the servers are up, then you can start installing the ESDT.

Installing ESDT's Using the Science Data Server GUI Continued



3 From **p0acs 03**, go to the directory, **/usr/ecs/TS1/CUSTOM/utilities**, type command:

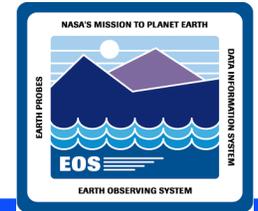
EcDsSdSrvGuiStart TS1

The Science Data Server GUI will be brought up. You will see the list of already installed ESDTs there. To add new one, click on the **Add** button. You will get the **Add ESDT** window.

This window is pointing to: **/usr/ecs/TS1/CUSTOM/data/ESS** where ESDT's have been staged.

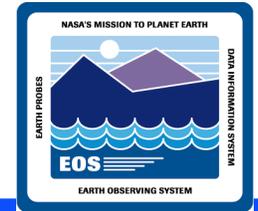
4 Click on the **File** button to select the descriptor file. Select the correct .desc file and click on **OK**. Wait and check the message at the bottom of the window. You will see message – **Successfully added...** Click the **Refresh** button and you will see recently added ESDT into the list.

Re-install or Update ESDT's Using the Science Data Server GUI



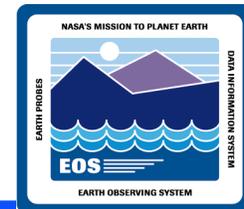
- Only the Science Data Server (SDSRV) needs to be brought down/up using the Kill and Start scripts before doing step 4 below.
- **Procedures:**
 - 1 telnet to (SDSRV) p0acs03[e.g.]**
 - 2 login:, password:**
 - 3 cd /usr/ecs/TS1/CUSTOM/utilities**
(Log into the (SDSRV) using **Maintenance Mode**. This locks out other users while The data base is being update with ESDT's.)
 - 4 Execute command : **EcDsScienceDataServerStart TS1 StartTemperature maintenance****
(This script will bring up the Science Data Server.)

Re-install or Update ESDT's Using the Science Data Server GUI Continued



- Then check the status of the SDSRV using a Server Monitor GUI WHAZZUP. Once all the servers are up, then you can start installing the ESDT.
- 5 From p0acs 03, go to the directory, /usr/ecs/TS1/CUSTOM/utilities, type command: EcDsSdSrvGuiStart TS1**
 - 6** The Science Data Server GUI will be brought up. You will see the list of already installed ESDTs there. To **Update** a ESDT, click on the Update button. You will get the **Update ESDT** window.
(This window is pointing to: /usr/ecs/TS1/CUSTOM/data/ESS where ESDT's have been staged.)
 - 7** Click on the **File** button to select the descriptor file. Select the correct .desc file and click on **OK**. Wait and check the message at the bottom of the window. You will see message – **Successfully added...** Click the **Refresh** button and you will see recently added ESDT into the list.
-

Science Data Server

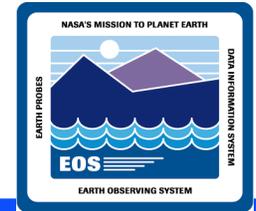


The Science Data Server Operator GUI provides the operator two major functions, the management of Earth Science Data Types and the management of all types of requests the Science Data Server Operator is involved with. Further details on these two functions are given in Table below.

Table. Common ECS Operator Functions Performed with the Science Data Server Operator GUI

| Operating Function | GUI | Description | When and Why to Use |
|--|----------------------------|--|---|
| Manage Science Data Server Earth Science Data Types (ESDTs) | Data Types Tab | Allows operators to manage the ESDTs offered by the Science Data Server | As needed, to manage data type descriptor information and add and update ESDTs |
| Manage Data Server System Requests | System Requests Tab | Allows operators to manage all the requests within each data server component | As required, to manage requests in each data server component |

Science Data Server Main Screen



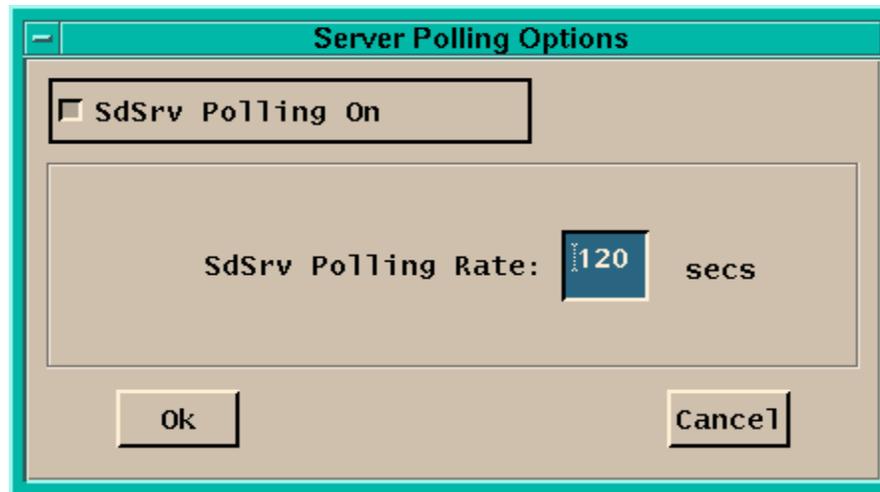
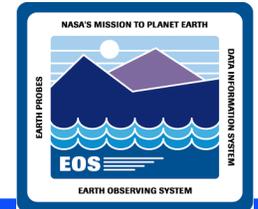
The ECS Science Data Server operator GUI, shown in slide/page 95, has two tabs that provide access to each one of the components' screens.

- The Earth Science Data Type Manager is accessed through the **Data Types** tab
- The System Request Manager is accessed through the **System Request** tab.

The operator can select from the menu bar items at the top of the Science Data Server Operator window for getting help and activating less-frequently used functions. The menu bar capability is available on all Science Data Server Operator GUI screens. The following menus are available:

- **File** - which includes the following item:
 - **Exit** (Ctrl-Q) - Exit application (graceful exit).
- **View** - functionality has not been determined as of this time (TBS).
- **Options** - This menu includes the *System Settings* item that opens the Server Polling Options window. Polling of the data server can be switched On/Off and the SdSrv Polling rate can be adjusted through this window shown in the next slide.
- **Help** - which provides context sensitive help.

Science Data Server Polling Options

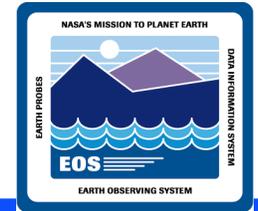


Science Data Server - Server Polling Options

Table. Science Data Server - Server Polling Field Description

| Field Name | Data Type | Size | Entry | Description |
|--------------|-----------|----------|----------|---|
| Polling Rate | integer | 4 digits | Optional | Specify the rate at which the Science Data Server Operator GUI is updated with data coming from the Data Server. The polling rate default is 120 seconds. |

Science Data Server Operator GUI Data Types Tab

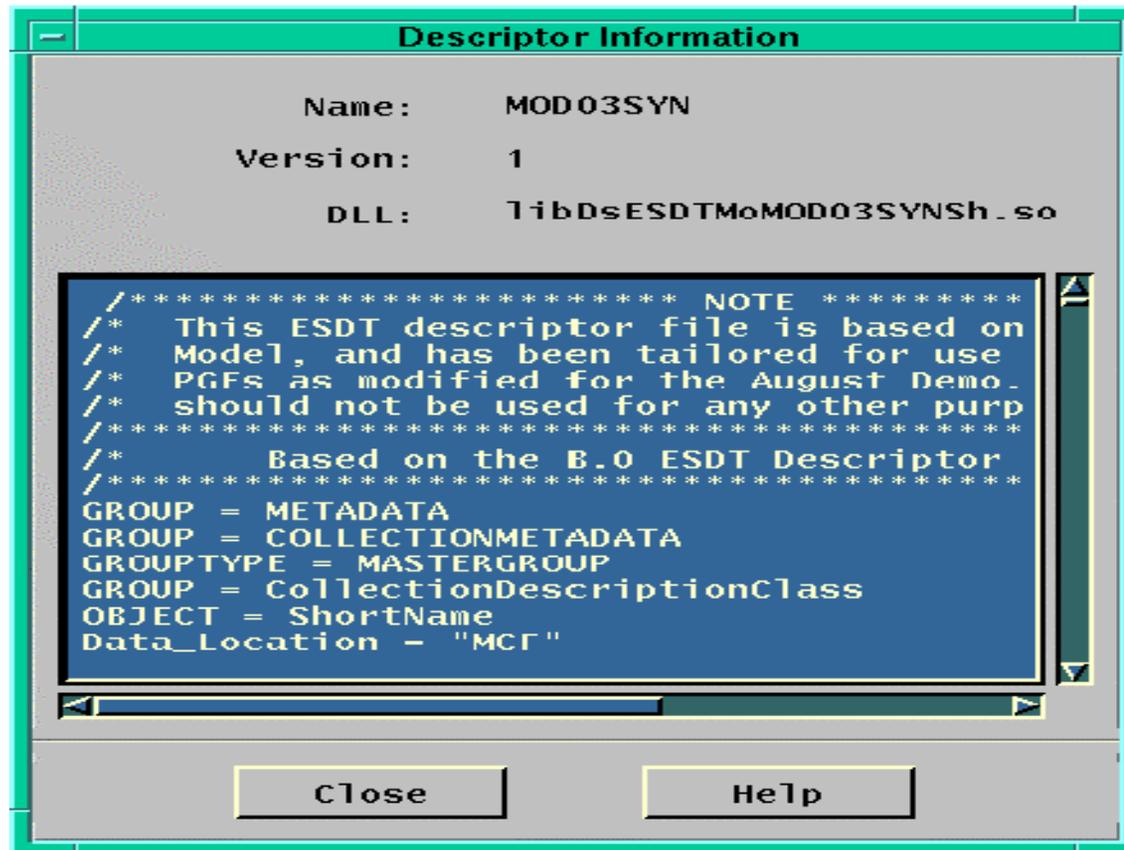
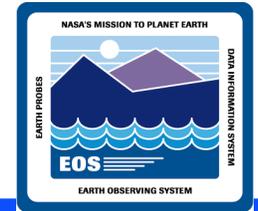


The Data Types Tab is the default screen of the Science Data Server Operator GUI shown in Slide/page 95. This window provides operations personnel at the DAAC the capability to view descriptor information, add new ESDTs and update ESDTs. A list of currently installed ESDTs is shown along with a version number and a brief description of the structure for an ESDT. Additional information that describes the structure, contents, and services that each existing ESDT provides can be viewed by selecting the data type and clicking on the *View* button.

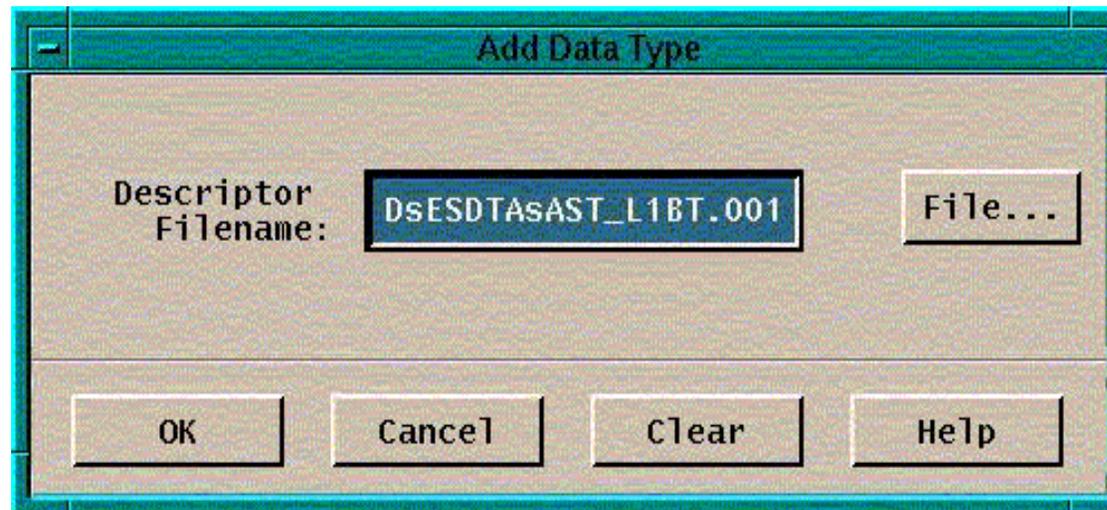
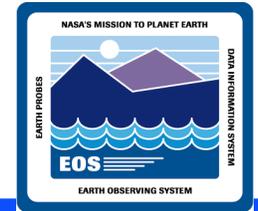
**Science Data Server Operator -
Data Types Field Description**

| Field Name | Data Type | Size | Entry | Description |
|--------------|-----------|------|------------------|---|
| Data Type ID | character | 8 | System generated | Uniquely identifies the specific type of ESDT. |
| Name | Character | 25 | System generated | Name of ESDT. |
| Version | Integer | 3 | System generated | Version number of ESDT, assigned starting at 1. |
| Description | Character | 255 | System generated | Includes structure and services available for an ESDT. |
| Find | Character | 255 | Optional | This functionality is provided in order to help the user browsing very long ESDT lists. |

Science Data Server Describer Information dialog



Science Data Server-Add Data Type Dialog



Science Data Server-Add Data Type Dialog Table

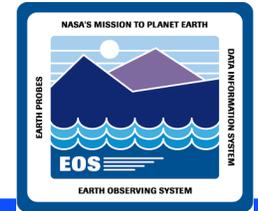
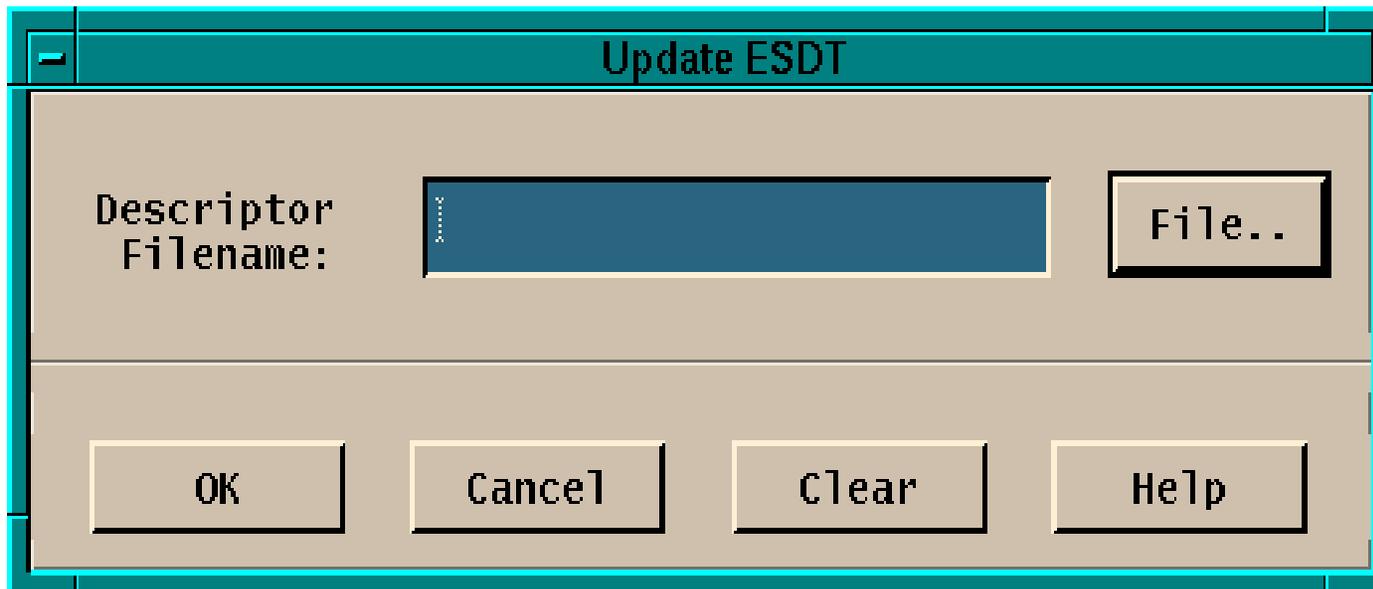
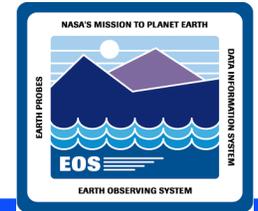


Table . Science Data Server - Add Data Type Field Description

| Field Name | Data Type | Size | Entry | Description |
|---------------------|------------------|------|----------|--|
| Descriptor Filename | Character string | 25 | required | Name of an ASCII file containing the ESDT descriptor file. |

Add ESDT Data GUI is the same as the Update ESDT GUI in its functionality.

Science Data Server Update ESDT GUI



Science Data Server - Update Data Type Field Description

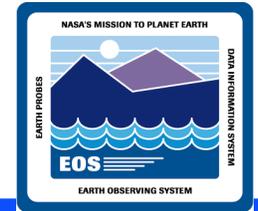
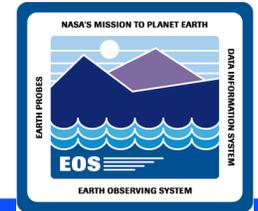


Table. Science Data Server - Update Data Type Field Description

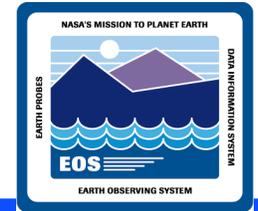
| Field Name | Data Type | Size | Entry | Description |
|---------------------|------------------|------|----------|--|
| Descriptor Filename | character string | 255 | required | Name of an ASCII file containing the ESDT descriptor file. |

- *Refresh/Reconnect* updates the data type information screen with current information.
- *Operator Messages* provides the functionality that displays informational and error messages to the user.

System Management Requests Window



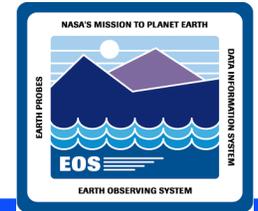
System Requests Tab



System Requests Tab

Clicking the System Requests tab will bring up the System Management Requests window. This window provides operations personnel at the DAAC the capability to monitor requests the Science Data Server is working with. All requests within the Science Data Server are displayed. The columns of the list can be sorted by positioning the cursor and clicking on the appropriate column of interest. The requests can be filtered by positioning the cursor and clicking on the **Filter** button and entering the attributes on which to filter.

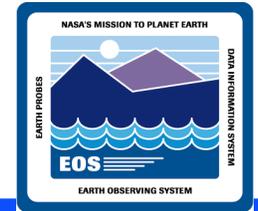
System Requests Tab Continued



In addition, the following buttons are provided:

- **Change Priority:** changes the priority of each selected request through a pulldown menu. Possible values are: Express, Very High, High, Normal (default) and Low.
- **Apply** allows the operator to commit to the priority change selected through the change priority button.
- **Filter...** brings up the System Management Filter Requests dialog which provides a selection of attributes on which to filter for the list of System-wide requests. Filter on system management requests by entering the desired information, then clicking on the Request ID or Requester radio button for the desired attribute.

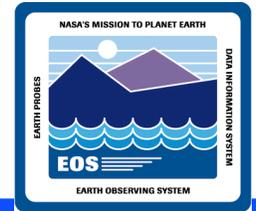
System Requests Tab Continued



- Return to the original list of requests by clicking on the All Requests radio button. Click on other filters associated with State and Priority by clicking on the toggle button. Filter on every attribute associated with a category by clicking the **All** button or clear a category of filters by clicking on the **None** button.

Memory State: it monitors the current memory state of the data server in regards to values that are set on the server side through configuration parameters. Possible values are: Normal(green color), Low(yellow), Very Low(red). This functionality will only be visible if the server's `DSSMEMORYMONITORDISABLEFLAG` is off.

System Management Filter Requests Dialog



System Management Filter Requests

System Management Filter Requests

Request ID

Requester

All Requests

State:

Submitted

Queued

Active

Complete

Failed

Priority:

Express

Very High

High

Normal

Low

OK Apply Cancel Help

System Management Filter Requests Dialog Continued

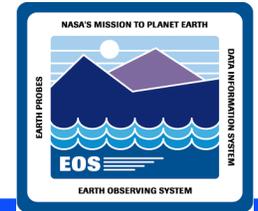


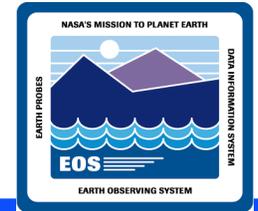
Table. System Management Filter Requests Field Description

| Field Name | Data Type | Size | Entry | Description |
|------------|--------------------|------|------------------|---|
| Request ID | character | 255 | system generated | Unique identifier for the request. |
| Requester | variable character | 100 | system generated | Identifies the user that submitted the request. |

In addition, the following buttons are provided:

- **OK** implements filter criteria, and the dialog closes.
- **Apply** implements filter criteria, and the dialog remains open for additional filtering.

System Management Filter Requests Dialog Continued

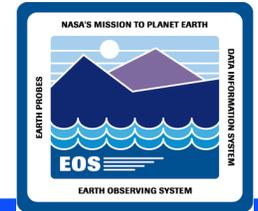


- **Cancel** closes the dialog without saving
- **Help** displays on-line help information.
- Back to the System Requests tab, **Operator Messages** provides informational and error messages to the user.
- **Refresh** causes the Data Server to be polled for an update on Requests.

Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to the 920-TDx-013 “Custom Code Configuration Parameters” documentation series . The “x” refers to the installed location, e.g. 920-TDG-013 is for GSFC DAAC

Database Interface



Databases

The Science Data Server Operator GUI does not include the direct managing of any database. It has an interface with the Science Data Server Data Base: however this interface is based on a simple parameter passing function. For further information of the Science Data Server Data Base refer to 311-CD-107-005, *Science Data Server Database Design and Schema Specifications for the ECS Project*.

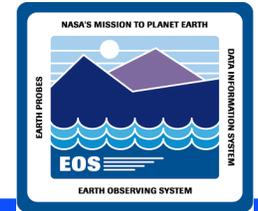
Special Constraints

The Science Data Server Operator GUI runs only if the Science Data Server is running in the background. Note also that at the moment the Science Data Server GUI is started through a command line that specifies the configuration file that is used to initialize the GUI Application.

Outputs

There is no processing associated with the operation of this GUI. The information provided to the operator are retrieved from the Data Server Database described in the previous sections and displayed through the screens discussed therein.

Production Rules Syntax Overview



PGE Registration

PGE Registration ODL format

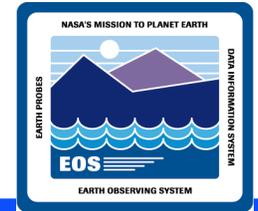
Production Rules

Rule Descriptions and Corresponding Syntax

Final Notes

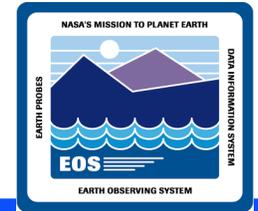
Science Software and Production Request

PGE Registration



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

PGE Registration ODL format



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

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

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